A DYNAMIC MODEL FOR DAYLIGHT IMPACT ON ENERGY SAVINGS IN DIRECT GAIN PASSIVE SOLAR BUILDINGS

Ahmed H. M. Mokhtar

A Thesis

in

The Centre for Building Studies

Faculty of Engineering and Computer Science

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Engineering (Building) at

Concordia University

Montreal, Quebec, Canada

September 1991

ISBN 0-315-68742-8

ABSTRACT

A DYNAMIC MODEL FOR DAYLIGHT IMPACT ON ENERGY SAVINGS IN DIRECT GAIN PASSIVE SOLAR BUILDINGS

Ahmed H. M. Mokhtar

Direct gain passive solar buildings are characterized by large glazing area which produces a considerable amount of daylight into their interiors. Currently, passive solar building designers are used to perceive daylight as a by-product of their designs. As an additional energy saving incentive, designers usually ignore the potentials of natural illumination while taking their design decisions. This thesis attempts to explore and assess the energy saving potential of daylight in direct gain passive solar buildings. In addition, this thesis examines the effect of daylight utilization on the design of fenestration. A mathematical model has been developed to provide a dynamic analysis for daylight energy benefits in passive solar buildings. The model simulates the time dependent nature of daylight and estimates indoor illumination levels. Furthermore, it quantifies the simultaneous solar energy gain within the space and employs these data to estimate the potential energy saving associated with daylighting. A case study incorporating a sensitivity analysis for window design variables has been examined. The results of this thesis have indicated that the energy advantages of direct gain passive solar buildings increase significantly when daylight energy benefits are considered. It also shows that considering daylight in the design process of the direct gain passive solar building will have an implication of their fenestration design.

ACKNOWLEDGEMENT

Thanks to GOD for every thing the author was able to achieve and for every thing he tried but failed to achieve.

The author also wishes to thank all the individuals whose assistance and encouragement were instrumental in the realization of this thesis. He is deeply grateful to his supervisor, Dr. Ramy El Diasty for his great help, guidance and useful suggestions which have improved the contents of this work. The author wishes to thank his co-supervisor, Dr. Paul Fazio for his constructive advices and thoughtful recommendations.

The author appreciates the financial assistance of Energy, Mines and Resources Canada and Natural Science and Engineering Research Council Canada that is provided through the operating research grants of professor El Diasty.

The continuous encouragement from Dr. Osama Moselhi was inspiring to the author in continuing this work, and the cooperation of Dr. Athienitis was helpful in the experimental part in this thesis.

The endless willing to help by Mr. Joseph Zilka in conducting the experimental work is highly appreciated.

The author is very grateful to his colleagues, Ismail Budaiwi, Tarek Hegazy, Adel Abdu for their valuable criticism, helpful advice and solid support.

Last but not least, very special thanks to the author parents Prof. Hanafy Mokhtar and Dr. Laila Zahran for their unlimited and unreplaceable giving.

TABLE OF CONTENTS

page

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

a_e	Room fenestration azimuth angle, *radian*
a_i	Incident angle between solar beam and a surface, *radian*
a_{ie}	Modified incident angle, *radian*
a_s	Solar azimuth angle from the south, *radian*
a_t	Solar altitude angle, *radian*
A_L	Area of the window's wall, m^2
A_r	Window area, m^2
A_s	Area covered by a reference point, m^2
A_w	Wall area, m^2
E_{dn}	Direct normal solar illuminance, *klux*
E_{kh}	Full sky illuminance on a horizontal surface, *klux*
E_{khs}	Horizontal illuminance from part of the sky facing the tilt surface, *klux*
E_{ks}	Illuminance due to part of the sky which does not face the surface, *klux*
E_{kt}	Sky illuminance on tilted surface, *klux*
E_{kv}	Sky illuminance on vertical surface, *klux*
E_P	Illuminance on a point, *klux*
E_r	Total daylight illuminance on a reference point, *klux*
E_{ref}	Illuminance due to interreflection between surfaces, *klux*
E_w	Illumination level required on the working plane, *klux*
E_{xt}	Extraterrestrial solar illuminance, *klux*
f_i	The ratio of the window's area that receives direct solar radiation.
G_T	Solar energy incident on a surface from clear sky, W/m^2
h_o	Coefficient of heat transfer of the outside surface air film, $W/(m^2.°k)$
I_b	Beam solar radiation, W/m^2
I_{bt}	Beam solar radiation on tilted surface, W/m^2
I_d	Diffuse solar radiation, W/m^2
I_t	Total solar radiation from clear sky at specific hour on a surface, W/m^2
I_d	Diffuse solar radiation from the sky on tilted surface, W/m^2

I_g Diffuse solar radiation from the ground on tilted surface, W/m^2

J Julian date.

l Latitude, *radian*

L Longitude, *radian*

L_{pc} Luminance of a point under clear sky, kcd/m^2

L_{po} Luminance of a point under overcast sky, kcd/m^2

L_{pp} Luminance of a point under partly cloudy sky, kcd/m^2

L_s Light source centroid luminance, kcd/m^2

L_{sp} Luminance of solar beam from a surface, kcd/m^2

L_{zc} Clear sky zenith luminance, kcd/m^2

L_{zo} Overcast sky zenith luminance, kcd/m^2

L_{zp} Partly cloudy sky zenith luminance, kcd/m^2

N_d Number of days per month.

P_{cs} The probability of the sky to be clear.

P_{os} The probability of the sky to be overcast.

P_{ps} The probability of the sky to be partly cloudy.

$q_{s,t}$ Heat gain at time t, W

Q Hourly solar radiation heating a room, W

Q_d Daily solar radiation heating a room, W

r_t Hourly total radiation over daily total radiation.

R_b The ratio between beam radiation on tilted and horizontal surfaces.

SM Time zone standard meridian.

ST Standard time in decimal hours.

t Solar time in decimal hours.

t_e Sol-air temperature, $°k$

t_o Out side temperature, $°k$

T_d Number of working days per month.

T_h Number of working hours per day.

U The difference between the window and the wall conductances, $W/(m^2.°k)$

U_L Wall conductance value, $W/(m^2.°k)$

U_w Heat gain and loss by conduction through the window glazing, *W/(m². °k)*

Greek Symbols

α Point azimuth angle from the sun, *radian*

β Absorbtion of the surface of the room.

δ Solar declination angle, *radian*

ε Omittance of a wall surface.

ζ Sky point altitude angle, *radian*

η Angle between the sun and the sky point, *radian*

θ Sky point zenith angle, *radian*

ρ_g Ground reflectivity.

ρ_{ave} Average reflectance of the room surface.

ψ Surface tilt angle on horizontal, *radian*

γ Surface - sun azimuth difference, *radian*

ξ Efficiency.

CHAPTER I

AN INTRODUCTION TO DAYLIGHT UTILIZATION IN PASSIVE SOLAR BUILDINGS

1.1. BACKGROUND ON PASSIVE SOLAR BUILDINGS

Climate sensitive design is not a new concept, its principles have been used by almost all indigenous builders for centuries. Only recently, since the industrial revolution, the talent of designing in harmony with nature has been lost. Mechanical power has become focus of interest in satisfying the human comfort needs. As a result of the limitation of conventional energy sources and its environmental pollution implications, governments and researchers have recently begun to search for new and renewable alternative energy sources.

During the last thirty years, solar energy utilization in buildings was among the most successful attempts to reduce dependency on fossil fuels. The sun continues to send out vast amount of energy. Although only one trillionth of this energy reaches the earth, this is still about 10,000 times the total commercial energy used [1], so that even very modest progress in the utilization of the solar energy could provide a large part of the energy needs. One of the currently most popular means for utilizing the sun's energy in buildings is passive solar design. It is based on the admission of the solar energy into buildings through south-facing glazed apertures. By doing so, free energy is utilized for space heating rather than resorting to elaborate and costly mechanical equipment. Passive solar design attempts to satisfy the building occupants' needs for space heating, cooling, and lighting as far as possible by natural means, using the building itself to collect, store, and distribute solar energy within the interior space.

The most common types of passive solar designs can be classified as direct gain systems, indirect gain systems, and isolated gain or sun space systems [2]. Direct gain passive system is the most popular type used for space heating. This type incorporates large south-facing glazed areas that permit significant amount of solar energy to be admitted and absorbed by the interior surfaces. The excess heat is stored in massive thermal storage which is strategically located to receive the admitted solar energy. During night or cloudy conditions, the heat stored is released from the thermal mass into the interior space.

1.2. RELATION BETWEEN PASSIVE SOLAR AND DAYLIGHT

Direct gain passive solar buildings are characterized by large glazing areas. As the sun penetrates the building through glass, it provides not only free heat but also free illumination. Sunlight acts as an abundant light source during daytime with a colour temperature that is pleasing to the human eye. If distributed uniformly, the lumens contained in a single square meter of sunlight could provide five hundred lux of illumination over an area of 180 m^2 [3]. When properly directed, the solar spot in an interior space acts as a very bright luminaire in that space. Such advantage is appreciated since electric lighting in commercial buildings consumes about 35% of the total building annual energy. In large buildings, electric lighting represents 50% of the total building energy consumption [3]. At the present time, energy benefits from the sunlight free illumination is rarely considered in passive building total energy savings. When it is assessed accurately, the attractiveness of the direct gain passive solar buildings in saving energy will be enhanced significantly.

1.3. IMPORTANCE OF THE TOPIC

Among the many researches realized in the passive solar buildings design, there had been little interest in daylighting energy benefits. One of the reasons is that the focus on passive solar application has been limited to the residential sector, and the potential for lighting as energy savings in that sector is minimal. As interest in the passive solar field extends to commercial buildings, electric lighting becomes a significant building energy consuming element, both directly and indirectly by virtue of its impact on cooling load. Although the breakdown varies considerably by building type and location, in office buildings for example, 50% of the building resource energy may be attributable to lighting [3]. In new buildings, as heating and cooling loads are reduced with the use of tighter building envelopes and improved HVAC design, lighting looms as the single largest energy consuming element in buildings. As Rosenfield and Selkowitz [3] estimated, 20% of the existing 2.4 billion m^2 of commercial floor space in the United States could be daylit and the electric lighting dimmed and turned off during portions of the day. The annual saving represented by daylighting these five hundred million m^2 of office space is estimated at 12.5 billion kWh. That potential in saving with daylight may be utilized in the passive solar buildings to increase their efficiency.

Several methods of assessment have been introduced for daylight design and others for passive solar design but none attempted to incorporate both. A model that incorporates both of solar heat gain and daylight benefits is essential to assess the combined heating-lighting saving potential. Such model needs to emphasize the effect of solar illuminance. Investigations are needed to explore the necessity to account for daylight in the energy analysis of passive solar buildings and to explore the potential for window design improvement for maximum heat gain and daylight utilization.

1.4. RESEARCH OBJECTIVES

The following are the three main objectives of this study:

i) Developing a dynamic daylight model for predicting the energy benefits of daylight in direct gain passive solar buildings, taking into account the illuminating effect of the admitted solar beam.

ii) Examining the role of the fenestration design in passive solar building in altering the combined benefits of solar heat and light.

iii) Providing passive solar building designers with a design tool in the form of a computer software that can predict daily and annual daylight profiles in buildings as well as the sizing of windows and their orientation in reducing energy consumption in buildings.

1.5. RESEARCH METHODOLOGY

To achieve the objectives of the research, the following eight step methodology has been employed.

i) Studying the existing methods used in designing passive solar building in general and daylight estimating methods in particular.

ii) Surveying existing literature in the field of daylighting.

iii) Analyzing the available daylighting computer models.

iv) Constructing a mathematical model for both solar heat gain and daylighting utilizing existing calculation procedures in both fields.

v) Coding the mathematical model into a computer program.

vi) Verifying the model and the computer program by simple field measurements.

vii) Applying model in a case study that incorporates generating a sensitivity analysis to different design possibilities of the window.

viii) Analyzing the obtained results and acquiring conclusions.

1.6. THESIS OVERVIEW

Each of the following chapters of this thesis is devoted to a specific aspect. In chapter II, a review and analysis of the existing literature related to daylight design is provided. Chapter III covers the design of the model showing the sequence used in its construction and provides insight into the equations utilized in the basic calculations. The skeleton of the computer code, its capabilities, limitations, and structure are presented in chapter IV along with its application. Chapter V describes the model validation and evaluation by through experimental tests. While a case study and the results of a sensitivity analysis constitutes the contents of chapter VI. Finally in chapter VII, summary of this thesis, the conclusions, and the contributions made along with suggestions for future work are presented. Appendix A contains the calculation procedure used to determine the sun spot position in a room, while appendix B presents a list of the computer program.

CHAPTER II

DESIGNING WITH DAYLIGHT: A REVIEW

Although the interest in daylighting is now increasing at a rapid pace, the fundamentals of the science has been established since the early decades of this century. The latest progress in computer technology, instrumentation and data acquisition coupled with a strong interest to conserve energy in buildings, have resulted in an impressive development in daylighting during the last two decades. The recent development is received here to constitute the necessary literatures review needed for this thesis.

2.1. AVAILABILITY OF DAYLIGHT

Daylight can be separated into direct sunlight and diffuse skylight. Illumination from direct sunlight has been investigated since the 1940s. Elvegard and Sjostedt [4] had published an equation to describe the solar illuminance in Stockholm, Sweden. Another equation was given by Chroscick [5]. Krockman and Seidl [6] had developed a model based on work done in Berlin, Gennany. The most widely used equation for direct sun light illuminance is the one recommended by the Illuminating Engineering Society of North America (IES) [7]. Daylight from the diffuse sky has always been studied through design skies, as actual skies are almost impossible to be numerically represented. Both clear and overcast design sky luminance distribution are well expressed by Kittler formula [8] and Moon and Spencer formula [9] successively. Yet other formulae can be found as those shown by Gillette, Pierpoint, and Treado [10]. For partly clouded design sky, its luminance distribution is given by the IES [7]. Another work was done by Tregenza [11] who calculated according to field measurements the association between global horizontal illuminance and cloud ratio in the partly cloudy sky at constant altitude. In another article [12], Tregenza described a model which requires simple meteorological measurements and amount of high, medium, and low clouds to give statistical data of the luminance of any given area of the sky. In another study [13], he also described a technique for calculating ground illuminance according to arbitrary cloudiness taking into account interreflections between cloud layers as well as that between the ground and the clouds. Diffuse sky calculation usually depends on the sky zenith luminance. Zenith luminance for different sky conditions were given by the IES [7] in the form of tables and equations. While Gillette and Kusuda [14] have provided formulae for calculating clear and overcast sky zenith luminance. El Diasty [15] studied the zenith luminance in Montreal during the different seasons of the year and derived equation for each season to get the zenith luminance value. Karayel, Navvab and Ne'eman [16] derived a new general equation for zenith luminance that can be applied up to latitude 76 degree and for a 1.5 to 8.5 turbidity range.

2.2. DAYLIGHT CALCULATION METHODS

Daylight calculation methods have been divided by Hopkinson [19] into two main groups:

i) Single stage methods where the total daylight is calculated by one operation based on a number of empirical assumptions and compromises. An example for that is the B.R.S daylight factor slide role.

ii) Stage by stage methods where the individual components of the daylight illumination reaching the reference point, that is the direct light, the reflected light from external and internal surfaces, are calculated independently and summed up to give the total daylight at the reference point. The stage by stage methods can be classified into the following:

- Tabular methods such as the N.P.L graded daylight factor tables and the B.R.S simplified daylight tables.
- Diagrammatic methods such as the B.R.S daylight protractor, the half cube method and the waldram diagram.
- Simulation methods where artificial skies and building models are used to estimate the daylighting in interior spaces.
- Mathematical methods like the IES daylight procedures and the lumen method.

In this review, the focus will be on the mathematical methods as they are the most accurate and comprehensive in calculating the daylight. One of these methods is the IES Daylight Procedure [19]. It is based on determining the quantity of illumination coming to the window surface, calculating the light flux actually entering the illuminated space, determining the distribution of the light flux within the room and calculating the illuminating flux produced on the surface of interest. Another method is the Lumen Method [20,21] which gives actual levels of illumination expressed in terms of maximum, minimum, and average values of daylight on the horizontal working plane. It is based on the coefficient of utilization concept and gives the illumination under uniform, CIE, and clear sky conditions. Different method was given by DiLaura [22] who described a technique for calculating equivalent sphere illuminance according to the direct component. The technique was initially oriented to artificial lighting calculations but DiLaura and Hauser [23] were able to extent the technique to include daylighting calculations but under constant sky conditions. The technique accounts for drapes, shades, and overhangs and has a shadow function. The components of daylight calculation were also subjected to improvement by many researchers. El Diasty, Shapiro and Fazio [24] developed calculations for the direct component that take into account the different sky conditions and the different sky luminance distribution through time. They also studied [25] both externally and internally reflected component of daylight under the same conditions. Nuretdinov K., Nuritdinov S. and others [26, 27] studied the effect of opposite building as an external reflecting surface on natural illumination and suggested a model for that purpose. DiLaura and others [22, 23, 28, 29] studied the internally reflected component of light including the case of unempty room.

2.3. ANALYSIS OF EXISTING DAYLIGHT COMPUTER MODELS

The computer models which were thought to be relevant to the course of this thesis are to be discussed. A survey was published by Lawrence Berkely Laboratory [30] to cover daylight computer programs available in U.S. up to June 1985. It compares the purpose of the different software, their capabilities, and their output features. The three most relevant models, Dalite, Superlite, and Jordaans' will be analyzed in the this section. By comparison with all models surveyed by the Lawrence Berkely Laboratory, the selected three models can be considered as most comprehensive. The purpose, capabilities, structure, output features, and limitations of each program will be discussed.

i) DALITE

Developed by Gillette and others. [References number 10, 14, 31]

- The purpose:

  A system of Fortran subroutines designed for inclusion into larger building energy simulation programs such as DOE-2 and BLAST. It makes it possible for these programs to account for energy trade offs associated with daylighting.

- Program capabilities:

  Dalite is capable to handle the following:

  - Windows, skylights, clear stories, roof monitors, overhangs, and vertical louvres.
  - Rooms having sloped ceilings.
  - Different sky conditions.
  - Direct beam illuminance as a rhomboidal patch of sunlight on the workplane.
  - Reflected light from surrounding buildings, ground, and internal surfaces.
  - Hour by hour calculations.

- Program Structure: (see Fig 2.1)

  The program consists of four main parts:

  - "DALITE" is the master subroutine which connect the program with the main program (e.g: DOE-2), by taking inputs and giving outputs.
  - "SKYLUM" is a subroutine for generating hourly exterior daylight values.
  - "RMLITE" is a subroutine which calculates the illuminance on a horizontal plane according to sky luminance, direct sun, external reflections, and internal reflections.
  - "LLOAD" is a subroutine for calculating the energy trade offs associated with the previously calculated illumination level.

- Output features:
    - Numerical results in a matrix form to be fed to the main program.
- Limitation:
    - Room plan should be rectangular.
    - Depth to hight ratio should be less than three to one.
    - Non-rectangular fenestration cannot be sufficiently represented.
    - Room is assumed to be empty.
    - All walls must be vertical.
    - Glare analysis is not included.
    - The illuminance effect of the solar beam on the reference point is calculated only if that point position is within the solar spot.

ii) SUPERLITE

Developed by Winkelmann, Selkowitz, and Modest.

[References number 32, 33, 34, 35]

- The purpose:

    To be integrated with the DOE-2 building energy analysis computer program for determining the impact of daylight utilization on heating and cooling loads, energy use, energy cost, and peak electrical demand.

- Program capabilities:
    - Accounts for site conditions (external obstructions, orientation,...etc.)
    - Variable window size, slope, and orientation.
    - Accounts for window management (response to solar gain, glare control, and

various light control strategies).

- Room shape can be irregular. (an important advantage)
- Relatively short computing time.
- Hour by hour simulation.

- Program structure: (see Fig 2.2)

There are three main stages:

- "Daylight factor preprocessor" which calculates the interior illuminance for a standard overcast sky and for clear sky with 20 different sun positions, for user selected room locations. Then these values are divided by the corresponding exterior illuminance to get factors which are stored for later interpolation.
- "Hourly daylighting simulation" which is done by interpolating the stored daylight factors using the current-hour sun position and cloud cover, then multiplying by the current-hour sun exterior horizontal illuminance.
- "Hourly lighting control" which simulating the stepped and continuously dimming lighting control systems to determine the electric lighting energy needed to make up the difference between the daylighting level and the design illuminance. Then determine hourly heating or cooling requirements for each space.

- Output features:

Series of reports that provide:

- Monthly and annual summary data.
- Hourly/monthly energy savings.
- Statistics on the frequency of occurrence of various interior daylight illuminance value for each reference point.

- Cumulative probability of exceeding certain illuminance value.
- Hourly values of daylight relating variables for particular day. (e.g: cloud amount, solar altitude, solar azimuth, external illuminance,....etc.)

iii) PROGRAM # 3 (Jordaans')

Developed by Jourdaans. [Reference number 36]

- The purpose:

  Calculating the total quantity of daylight provided to an arbitrary position in a room.

- Program capabilities:
  - Considers direct and reflected daylight (opposite buildings, ground and interreflected).
  - Computes the shadow patterns of the surrounding buildings for the cloudless sky.
  - Can compute the influence of reflectance from surfaces whose colour differ from that of the wall (such as doors).
  - Possible to calculate illuminance provided by artificial lighting.
- Program structure: (Not found in literature).
- Output features:
  - Illuminance due to direct incident daylight on the walls, floor, and ceiling.
  - Illuminance on each reference point due to:
    - only direct incident daylight.
    - both direct and indirect daylighting.

- Program limitations:
  - Does not include direct sun.
  - Assumes that all light is reflected diffusely from all surfaces.
  - Illuminance is not distributed gradually on the surfaces but each surface is divided into rectangles with equal illuminance.

The previous analysis has shown the variety of approaches to construct a daylight program. The three analyzed programs have been designed differently so as to achieve their purposes.

2.4. ENERGY SAVING POTENTIAL FROM DAYLIGHT UTILIZATION

Daylight potential for energy saving has become a point of interest to many researchers. Literature focusing on this topic differ in their investigating methods. Some have used very simple ways and others went through complex processes. A simple way was done by Bobenhausen and Lewis [37] who investigated an office building model and assumed a minimum daylight penetration through the depth of the perimeter zone of the building equal to 2% of the quantity of daylight falling on the respective wall surface and from that they calculated the reduction in energy if the artificial light turned off. The BRE [38] used more advanced method that predicts the saved energy by using probability charts for switching off artificial lighting which were derived from previous BRE behavioral studies. The prediction did not account for reduction in cooling load associated with daylighting utilization. Another prediction was done by Sanchez and Rudoy [39], they used simple computer program (which assumes uniform sky condition) to calculate hour by hour daylight availability and the energy associated with electric lighting and the annual cooling and heating loads. Oatman and Robbins [40] evaluated eleven passive solar strategies (including five utilizing daylighting) by studying their impact on energy consumption using an energy analysis computer program linked to a daylight computer program. Daylight calculations were made on hourly basis and consequent energy reduction was then estimated. Leaver and McQueen [41] used the PC-DOE microcomputer program in addition to DAYLITE program and field measurements to investigate the reduction in electric lighting and cooling loads associated with daylighting utilization in Louisiana. An advanced energy simulation program DOE-2.1B (which account for daylighting energy impacts) was used by Gates and Wilcox [42] to analyze the effects of different sizes of unilateral opening, clear story, and top lighting for three different climatic zones in California on the total energy consumption in classrooms with different orientations. Johnson, Sullivan, Selkowitz and others [43] have studied the influence of glazing systems (including orientation, window area, glazing properties, window management strategy, installed lighting power, and lighting control strategy) on annual energy use in prototypical office buildings. They used DOE-2.1B energy simulation program which has made the investigation of different parameters that are governing the fenestration design becomes possible. The previous investigation has indicated that daylighting utilization, specially in medium size non-domestic buildings, can result in reducing the lighting electricity up to 90% and decreasing the total energy consumption up to 30%.

2.5. SUMMARY

From the literature review, the main points that are important to the core of this thesis are summarized as follows :

- There is no single comprehensive procedures for daylight calculation that is widely accepted, hence there are several available channels to construct a calculation sequence for daylight prediction that would satisfy any specific requirements.
- Different models have utilized different methods so as to serve their design purpose.
- The IES daylight procedures can be the best available way to serve as a base for constructing the required model as it allows calculating the availability of daylight at a specific location in a room through the different sky conditions.
- Some models are more recognized than others and their utilization becomes more reliable.
- It is expected that utilizing daylight in direct gain passive solar buildings will have valuable reduction in their energy consumption.

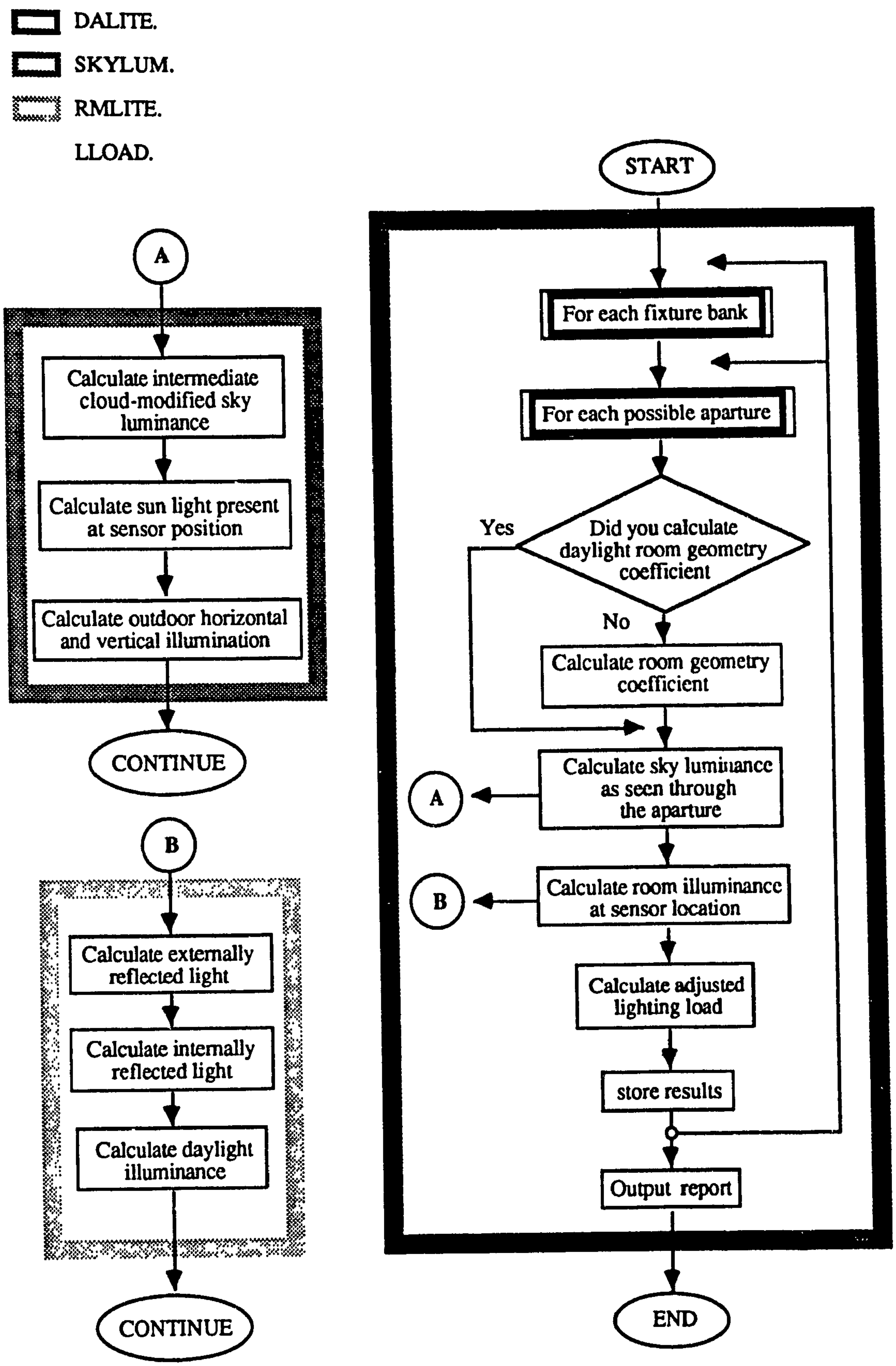


Fig 2.1 Flow Chart For "DALITE" Computer Program

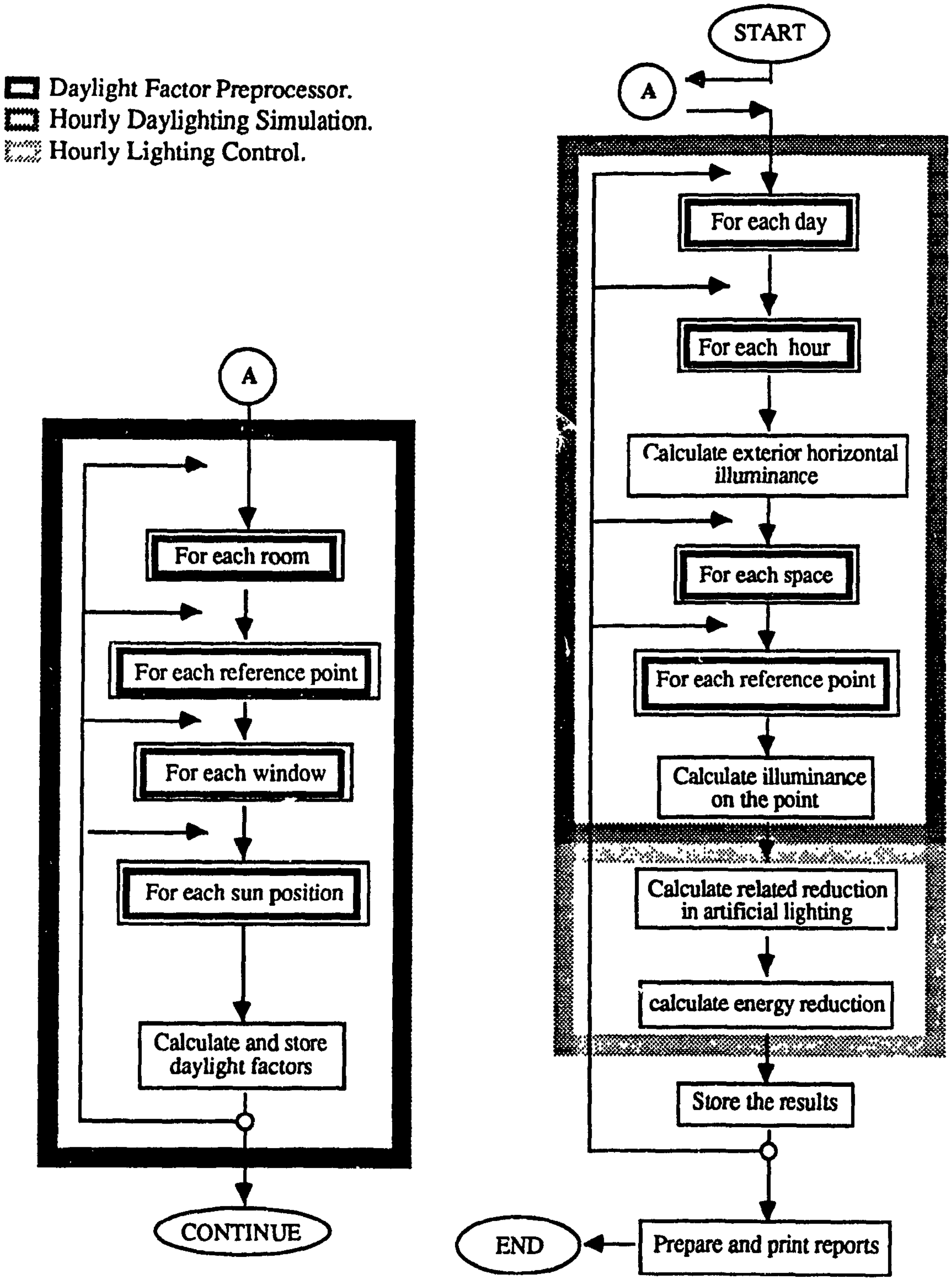


Fig. 2.2 Flow Chart For "SUPERLITE" Computer Program

CHAPTER III

MODEL DEVELOPMENT AND COMPUTATIONAL PROCEDURES

3.1. SEQUENCE FOR THE MODEL STRUCTURE

The proposed mathematical model consists of four main parts. In the first part natural light indoor, under different sky conditions and through a given window, is quantified. The effect of window configuration on the illumination level of a certain point in the room is then assessed. The amount of direct solar gain received in a room through that window is then calculated. Values obtained are then utilized to estimate the combined potential energy saving of both daylight and solar energy.

3.2. QUANTIFYING THE AVAILABLE DAYLIGHT

Daylight is available through its two components, the direct sunlight and the diffuse skylight. Both are dependent on the sun's position in the sky. In this part, the sun position will be calculated then the available direct and diffuse light on surfaces with various azimuth angles and in different latitudes will be determined.

3.2.1. Determination of sun position

Determining the sun position requires knowing the solar altitude, $\boldsymbol{a_t}$ and the solar azimuth, $\boldsymbol{a_z}$ (See Fig.3.1) which can be defined by knowing the site latitude, $\boldsymbol{l}$, Julian date, $\boldsymbol{J}$ (See Table 3.1) and time of day. Solar time of day is obtained as a function of Time zone standard meridian, $\boldsymbol{SM}$ and the standard time at which the sun position needs to be determined, ***ST***. The solar time is first calculated as a function of the standard time using the equation of time given by the IES [7].

$$ET = 0.170 \sin\left[\frac{4\pi}{373}(J-80)\right] - 0.129 \sin\left[\frac{2\pi}{355}(J-8)\right] \quad (3.1)$$

where ***ET*** = equation of time (decimal hours).

and the Solar time ***t*** can be calculated by [7]:

$$t = ST + ET + 12(SM-L)/\pi \quad (3.2)$$

The solar declination angle, δ, which accounts for the change of the earth's axis tilt with respect to its ecliptic orbit throughout the year can be determined as [7].

$$\delta = 0.4093 \sin\left[\frac{2\pi}{368}(J-81)\right] \quad (3.3)$$

Then, the solar altitude a_t is calculated as [7]:

$$a_t = \sin^{-1}(\sin l \sin\delta - \sin l \sin\delta \cos \pi t/12) \quad (3.4)$$

This equation gives a negative value if the sun is below horizon.

The solar azimuth measured from due south, a_s is determined from these equations which were developed from those given by kittler [44]

- If $t \leq 12$ then

$$a_s = \left[\cos^{-1}\left(\frac{\cos l \sin\delta + \sin l \cos\delta \cos(\pi t/12)}{\cos a_t}\right)\right] - \pi \quad (3.5)$$

- If $t > 12$ then

$$a_s = \pi - \cos^{-1}\left(\frac{\cos l \sin\delta + \sin l \cos\delta \cos(\pi t/12)}{\cos a_t}\right) \quad (3.6)$$

After determining the sun position, it will be used in calculating both of the diffuse sky illuminance and the direct solar illuminance.

3.2.2. Diffuse sky illuminance

Diffuse sky illuminance received on a surface can be estimated as a function of the sky luminance. The sky luminance depends on the sky condition which is typically classified as clear, partly cloudy or overcast. The clear sky has less than 30% cloud cover while the partly cloudy sky has between 30% to 70% cloud cover, otherwise the sky is considered to be overcast [45].

The luminance of a point within a clear sky dome depends on the clear sky zenith luminance, L_{zc} which can be determined by the equation of Gillette and Kusuda [14]

$$L_{zc} = (514 + 3611\ a_t^2)/1000 \qquad (kcd/m^2) \tag{3.7}$$

where a_t is in radian

Then, the clear sky point luminance, L_{pc} can be determined by Kittler's equation [8]

$$L_{pc} = L_{zc}\left[\frac{[\ 0.91 + 10\ \exp(-3\eta) + 0.45\ \cos^2\eta\ ]\ [\ 1 - \exp(-0.32/\sin\zeta\ ]}{0.274\ [\ 0.91 + 10\ \exp(-3\ (\frac{\pi}{2} - a_t)) + 0.45\ \sin^2 a_t\ ]}\right] \tag{3.8}$$

Where ζ is the altitude angle of the sky point.

η is the angle between the sun and the sky point. (see Fig 3.2)

η can be calculated using the following equation [7]

$$\eta = \cos^{-1}\ [\sin a_t \sin\zeta + \cos a_t \cos\zeta \cos\alpha] \tag{3.9}$$

where α is the point azimuth angle from the sun (Fig.3.2)

Similarly the partly cloudy sky zenith luminance, L_{zp}, which was driven from a table given by IES [7] should be known

$$L_{zp} = 0.683 - 0.3744\, a_t + 0.744\, a_t^2 \qquad (Kcd/m^2) \qquad (3.10)$$

Then the luminance at any point in the partly cloudy sky can be estimated using the equation given by Pierpoint [46]

$$L_{pp} = L_{zp} \left[\frac{[\, 0.526 + 5 \exp(-1.5\eta) \,]\, [\, 1 - \exp(-0.8/\sin\zeta) \,]}{0.551\, [\, 0.526 + 5 \exp(-1.5\, (\frac{\pi}{2} - a_t)) \,]} \right] \qquad (3.11)$$

Overcast sky zenith luminance, L_{zo}, can be calculated by equations recommended by IES [7].

$$L_{zo} = \frac{1.286}{\pi} (0.3 + 21 \sin a_t) \qquad (kcd/m^2) \qquad (3.12)$$

another equation, which is used in the model, is given by Gillette and Kusuda [14]

$$L_{zo} = (123 + 8600 \sin a_t) / 1000 \qquad (3.13)$$

Then the luminance at any point can be estimated by [14]

$$L_{po} = \frac{L_{zo}}{3} (1 + 2 \sin\zeta) \qquad (3.14)$$

For calculating the illumination from the sky on a surface, integration should be done to the luminance of the area in the sky dome which will face that surface. The surface possible positions can be vertical, horizontal, or tilted.

i) <u>*Sky illuminance on a vertical surface*</u> (E_{kv})

The following integration calculates the sky illuminance on a vertical surface [7]

$$E_{kv} = \frac{1}{\pi} \int_{\gamma - \frac{\pi}{2}}^{\gamma + \frac{\pi}{2}} \int_{0}^{\frac{\pi}{2}} L_p \sin^2\theta \sin\alpha \, d\theta \, d\alpha \qquad (3.15)$$

where γ = solar — elevation azimuth (Fig 3.3)

$\theta = \pi/2 - \zeta$

$L_p = L_{pc}$ for clear sky.

$= L_{pp}$ for partly cloudy sky.

$= L_{po}$ for overcast sky.

Equation (3.15) can be put in the following fom to be suitable for computer programming [47]

$$E_{kv} = \sum_{\theta=0}^{\pi/2} \left[\left(\frac{\theta_2 - \theta_1}{2} + \frac{\sin 2\theta_1 - \sin 2\theta_2}{4} \right) \sum_{\alpha=\gamma-\frac{\pi}{2}}^{\gamma+\frac{\pi}{2}} L_p \left| (\cos\alpha_1 - \cos\alpha_2) \right| \right] \qquad (3.16)$$

where θ_1, θ_2, α_1, α_2 are the parameters for the sky element as in Fig 3.4.

ii) <u>*Sky illuminance on a horizontal surface*</u> (E_{kh})

The illuminance on a horizontal surface can be calculated using the following equation [7]

$$E_{kh} = \frac{1}{\pi} \int_{0}^{2\pi} \int_{0}^{\frac{\pi}{2}} L_p \sin\theta \cos\theta \, d\theta \, d\alpha \qquad (3.17)$$

which can also be represented in the form given by [47]

$$E_{kh} = \sum_{\theta=0}^{\frac{\pi}{2}} \left[\left(\frac{\sin^2\theta_2 - \sin^2\theta_1}{2} \right) \sum_{\alpha=0}^{2\pi} L_p \ (\alpha_2 - \alpha_1) \right] \qquad (3.18)$$

Other simpler relations for E_{kh}, which are used in the model, are given by IES [7] as:

A) For clear sky:

$$E_{kh} = 0.8 + 15.5 \sin^{0.5} a_t \qquad (3.19)$$

B) For partly cloudy sky:

$$E_{kh} = 0.3 + 45 \sin a_t \qquad (3.20)$$

C) For overcast sky:

$$E_{kh} = 0.3 + 21 \sin a_t \qquad (3.21)$$

iii) *Sky illuminance on a tilted surface* (E_{kt})

Illuminance on a tilted surface is expressed as follows [47]:

$$E_{kt} = E_{kht} \cos\psi + E_{kv} \sin\psi \qquad (3.22)$$

where ψ is the angle between the surface and a horizontal plane (Fig.3.5).

E_{kht} is the horizontal illuminance from part of the sky dome facing the surface.

$$E_{kht} = E_{kh} - E_{ks} \qquad (3.23)$$

where E_{ks} is the illuminance due to part of the sky dome which does not face the surface and can be determined by the equation [47]

$$E_{ks} = \sum_{\theta=\frac{\pi}{2}-\psi}^{\frac{\pi}{2}} \left[\left(\frac{\sin^2\theta_2 - \sin^2\theta_1}{2} \right) \sum_{\alpha=\frac{\pi}{2}-\gamma+\phi_s}^{\frac{3\pi}{2}-\gamma-\phi_s} L_p \; (\alpha_2-\alpha_1) \right] \tag{3.24}$$

where

$$\phi_s = \sin^{-1}\,(\cot\theta_{avrg}\,\cot\psi) \tag{3.25}$$

Using the previous equations, it is possible to calculate the sky illuminance on a space fenestration and consequently determine its luminance which can be used to get the illuminance level in the space due to sky dome.

3.2.3. Direct solar illuminance

For a given position of the sun in the sky, the solar illuminance on a surface which has an azimuth difference angle with the sun, δ can be assessed as long as the absolute value of δ is less than 90. Procedures for this assessment start with determining the extraterrestrial solar illuminance, E_{xt} given by [7]

$$E_{xt} = 127.5 \left[1 + 0.034 \cos \left(\frac{2\pi}{365}\;(J - 2)\right)\right] \qquad (Klx) \tag{3.26}$$

and the direct normal solar illuminance is given as [7]

$$E_{dn} = E_{xt}\,\exp^{(-c/\sin\alpha_t)} \tag{3.27}$$

where c = 0.21 for clear sky.

c = 0.80 for partly cloudy sky.

note that $E_{dn} = 0$ for overcast sky.

The calculation of the angle a_i between the beam radiation and the surfaces on which there is a sun spot is made using the equation:

$$a_i = \cos^{-1} (\cos a_t \cos\gamma \cos\lambda + \sin a_t \sin\lambda) \qquad (3.28)$$

Where γ = surface - sun azimuth difference.

λ = surface tilted angle from vertical.

Then, calculation of the luminance of a surface that is produced by the solar beam, L_{sb} is made as follows;

$$L_{sb} = E_{dn} \cdot \cos a_i \qquad (3.29)$$

3.3. ESTIMATING ILLUMINANCE INSIDE A ROOM

To predict the energy saving due to daylight in a room, the illuminance level on the room's work plane at a sufficient number of reference points should be determined. Assuming an unobstructed window, each reference point will receive direct light from the window and multiple reflected light from the room surfaces (walls, ceiling, and floor). Direct light through windows includes the solar beam illumination. In this model, the illuminance from the direct solar beam on a reference point will be ignored as it will result in a very high illuminance level which may not be a true representation throughout the work plane. The present model considers the sunspot which results from the penetration of the solar beam through the window as a direct source of light illuminating the room surfaces only. Then these surfaces will reflect the light to the reference points of evaluation. The total illuminance at the reference points is induced by windows and room surfaces luminance. The luminance of a room surface is the product of the surface reflectance and its total illuminance. Total illuminance on a surface is estimated as follows: First the sunspot location in the room is determined, then the illuminance, on each room surface, due to the primary sources (window and solar spot) is determined. Accordingly, the calculation of luminance due to interreflection between the interior surfaces is determined and the total illuminance on all specific reference points is estimated. To reduce the calculation time, each room surface will be divided into three parallel strips and the luminance of the center point of each strip represents the average luminance on that strip.

3.3.1. Sunspot location in the room

The position of the sun spot within the room is determined at any given time. The calculation procedure accounts for tilted windows which are usually featured in the passive solar buildings. The procedure is designed for rectangular rooms with windows on vertical or tilted walls.

After determining the solar spot position, it is approximated by a rectangular shape in order to make illumination calculations simpler and faster. A prespecified 15 different locations for the spot position in the room are determined. The four corners of a window are denoted as 1,2,3,4 and the projected corners of the solar spot due to that window are denoted as 1',2',3',4' (see Fig. 3.6). Each corner point of the solar spot can be either on the floor, on the side wall or on the back wall. Table (3.2) and Fig. (3.7) to Fig. (3.21) show all the possible combination of the solar spot corners location. To identify which case is most applicable, the following nine parameters are determined, assuming the room geometry shown in Fig. (3.6), using the following variables:

ψ = window tilt angle.

γ = Window - sun azimuth difference.

ε = The projection of the solar altitude angle, a_t, on a plane vertical to the window.

$$\text{where} \quad \varepsilon = \tan^{-1} (\tan a_t / \cos \gamma) \qquad (3.30)$$

The nine parameters needed to identify the most applicable case are presented in the following table:

$x1 = length - (m+x)$	$s = a + b$	$s1 = a' + b'$
$z = b \tan \gamma$	$z' = b' \tan \gamma$	$P1 = a' + x1 / \tan \gamma$
$P2 = a + x1 / \tan \gamma$	$P3 = a + (m+x1) / \tan \gamma$	$P4 = a' + (m+x1) / \tan \gamma$

See Fig. 3.6.

After determining the values of these parameters, the flow chart in Fig. 3.22 is used to determine the case number of the sun spot position in the room. Detailed calculations for the determination of the final position and shape of the sun spot that is used in the illumination calculations are shown in Appendix A.

3.3.2. Illuminance on the room surfaces

The windows and the sun spots in the room are considered to be the primary sources of daylight for the room surfaces. Assuming that the window glazing is a perfect diffuser, the luminance of a window is determined by multiplying the transmittance of the window glass by the diffuse sky illumination received by the window. Similarly, the sun spot luminance from a surface is the product of the solar beam illuminance on the surface and the surface reflectance.

There are three possibilities for the relation between the light source and the point at which light is estimated. The light source may be either parallel, perpendicular, or tilted. The following equations [48] are used in assessing the illumination at any given point:

- *From parallel source*

$$E_p = \frac{L_s}{2}\left[\frac{f-a}{\sqrt{c^2+(a-f)^2}}\tan^{-1}\left(\frac{m}{\sqrt{c^2+(a-f)^2}}\right)\right] \quad (3.30)$$

- *From perpendicular source*

$$E_p = \frac{L_s}{2}\left[\tan^{-1}\frac{m}{a} - \left(\frac{a}{\sqrt{a^2+f^2}}\right)\tan^{-1}\left(\frac{m}{\sqrt{a^2+f^2}}\right)\right] \quad (3.32)$$

- *From tilted source*

$$E_p = \frac{1}{2\pi}\left[\begin{array}{c}\tan^{-1}\frac{m}{a} + \frac{f\cos\psi - a}{G}\tan^{-1}\frac{m}{G} + \\ \frac{m\cos\psi}{H}\left(\tan^{-1}\frac{f-a\cos\psi}{H} + \tan^{-1}\frac{a\cos\psi}{H}\right)\end{array}\right] \quad (3.33)$$

Where

E_p = Illuminance at the point.

L_s = Source centroid luminance.

and variables G and H are estimated as:

$$G = \sqrt{f^2 + a^2 - 2af\cos\psi} \qquad H = \sqrt{m^2 + a^2\sin^2\psi}$$

The definition of the parameters a, m, f, x, y, z, ψ are shown in Fig. (3.23).

Using these relations, the illuminance on each room surface due to both window and the sun spot can be calculated and added up for each surface. Consequently, each surface luminance can be determined by multiplying the illuminance value by the surface reflectance. The surface luminance will be used to calculate the interreflections between the surfaces.

3.3.3. Interreflection within the room

The room space is represented by six nodes, each node represents a surface. Assuming that the room acts as an ulbricht integrating sphere [19], which is a valid assumption for wall reflectance approaching the ceiling reflectance and room depth less than twice the ceiling hight, the room interreflected illuminance is obtained through the following equation [31]

$$E_{ref} = \frac{\sum_{n=1}^{6} E_n A_n}{(1-\rho_{ave})\, A_t} \qquad (3.34)$$

Where E_n = Total luminance on the surface #n from the window and the solar spot.

A_n = Area of the surface #n.

A_t = Total area of the surfaces.

and ρ_{ave}, the average reflectance of the room surfaces is determined as

$$\rho_{ave} = \frac{\sum_{n=1}^{6} \rho_n \cdot A_n}{A_t} \qquad (3.35)$$

The illuminance due to interreflection is then multiplied by each surface reflectance and added up to the surface luminance due to the window and the solar spot to obtain the total luminance on the surface.

3.3.4. Illuminance level at the reference point

The final illuminance level at the reference point is attributed to the direct luminance from the window and the luminance of room surfaces. Equations 3.31, 3.32, 3.33 will be used to calculate the illuminance at each reference point. If the calculated daylight illuminance at any reference point is equal to or exceeds the required illuminance, the electric light is shut off using automatic control system.

3.4. DETERMINING THE DIRECT SOLAR GAIN THROUGH A WINDOW

The recommended procedures by ASHRAE has been employed in this section to determine the hourly available solar energy. These procedures provide the means to accommodate windows having arbitrary orientation, overhangs, and side fins. The model has also utilized the method given by Duffie and Beckman [50] to accurately calculate the transmission and the utilization of the available solar energy through a room window.

3.4.1. Hourly available solar radiation.

An hour by hour evaluation is necessary in order to estimate the energy saving during variable daytime conditions as will be shown later in section 3.5. A surface such as a room window can receive hourly total rate of solar radiation according to the following equation [49]:

$$I_t = I_b + I_d + I_g \qquad [W/m^2] \qquad (3.36)$$

Where

I_t = Hourly total rate of solar radiation on the surface.

I_b = Hourly total rate of direct beam radiation.

I_d = Hourly total rate of diffuse sky radiation.

I_g = Hourly total rate of reflected radiation from the ground.

The direct beam radiation at window surface is defined as:

$$I_b = I_{DN} \, . \cos a_i \qquad [W/m^2] \qquad (3.37)$$

Where I_{DN} is the direct normal radiation given by

$$I_{DN} = A \;/\; e^{B/\sin a_t} \qquad [W/m^2] \qquad (3.37a)$$

Where the values A and B are given in Table 3.3.

The diffuse sky radiation I_d can be calculated using the equation

$$I_d = C \, . \, I_{DN} \, . \, (1 + \cos \psi) \,/\, 2 \qquad [W/m^2] \qquad (3.38)$$

Where C is the diffuse radiation factor given in Table 3.3 and

ψ is the tilt angle of the surface from the horizontal.

The reflected radiation from ground I_g is given by

$$I_g = I_{DN} \, . \, (C + \sin a_t) \, . \, \rho_g \, . \, (1 - \cos \psi) \,/\, 2 \qquad [W/m^2] \qquad (3.39)$$

Where ρ_g is the ground reflectance.

3.4.2 Hourly solar heat gain

As the available solar radiation on a room window is determined, it is possible to know the solar heat gain in the room. To consider the reduction in the transmitted radiation through the window due to glass reflection and absorption, both reflected and absorbed portions of the beam and diffuse radiation are determined. The transmittance of radiation, when only reflection is considered, (τ_r) is given by [50]

$$\tau_r = \frac{1}{2}\left[\frac{1-r_{\parallel}}{1+r_{\parallel}} + \frac{1-r_{\perp}}{1-r_{\perp}}\right] \qquad (3.40)$$

While the transmittance if only absorbtion is considered, (τ_a)is given by [50]

$$\tau_a = e^{-KL/\cos\theta_2} \qquad (3.50)$$

Where $r_{\parallel}$ is the parallel component of the radiation.

$r_{\perp}$ is the perpendicular component of the radiation.

$$r_{\parallel} = \frac{\tan^2(\theta_2 - a_i)}{\tan^2(\theta_2 + a_i)} \qquad (3.42)$$

$$r_{\perp} = \frac{\sin^2(\theta_2 - a_i)}{\sin^2(\theta_2 + a_i)} \qquad (3.43)$$

Fig [3.24] shows the angle of refraction (θ_2) which is given by [50]:

$$\theta_2 = \sin^{-1}\left[\frac{\sin a_i}{1.526}\right] \qquad (3.44)$$

The variable *(K)* is the extinction coefficient which is a property of the utilized glass having the unit (m^{-1}), and the variable *(L)* is the thickness of the glass in (m). Finally the portion of the beam solar radiation that is transmitted to the space (τ) is

$$\tau \cong \tau_r \cdot \tau_a \tag{3.45}$$

The preceding procedure is applicable only to beam radiation but it could be used for diffuse radiation if some modification in the incident angle, a_i is made as suggested by Brandemuel and Beckman [51]:

- For radiation from the ground

$$a_{i_e} = 1.043 + 0.0938\,\psi + 0.1543\,\psi^2 \tag{3.46}$$

- For radiation from the sky

$$a_{i_e} = 1.043 - 0.1307\,\psi + 0.08577\,\psi^2 \tag{3.47}$$

So by replacing a_i by a_{i_e} from the first equation, the transmittance of the radiation reflected by the ground, τ_g can be obtained, and by a_{i_e} from the second equation, the sky radiation equivalent transmittance, τ_d can be calculated. Thus the general equation governing the hourly radiation that heats up the room becomes

$$Q = A_r \cdot \beta \cdot (f_i \cdot \tau \cdot I_b + \tau_d \cdot I_d + \tau_g \cdot I_g) \quad \text{[Watt]} \tag{3.48}$$

where A_r is the window area.

β is the effective absorptance of the room.

f_i is the ratio of the window that receives direct beam radiation to the total window area.

The effective absorptance, β, is expressed as [50]

$$\beta = (1 - \rho_R) / (1 - \rho_R + \rho_R \cdot \tau_d \cdot A_r / A_R) \tag{3.49}$$

where ρ_R is the reflectance of the room surfaces.

A_r is the window area.

A_R is the room surface area.

The area of the window that receives direct beam radiation can be calculated using the method given by Sun [52].

3.5. ENERGY SAVINGS ESTIMATION

In the previous sections, a procedure for the availability of both daylight and solar energy in a room has been developed. A simple, direct procedure, which provides quantitative energy saving assessment that puts the use of daylight and solar energy in a single form, is suggested. This procedure will be demonstrated by a sensitivity analysis for window variables.

The proposed energy savings estimation procedure is a further development to an existing procedures by Rundquist [53]. The proposed procedure accounts for visible light, solar heat gain, and the thermal conductance as the window energy parameters. Among the advantages of the present method over Rundquist, is the fact that the present method takes into account the infiltration losses as well as it provides detailed calculations for the direct solar beam illuminance, as an input data rather than using the approximate daylight calculations given by Rundquist that ignores the sun illuminating effect. Also, the present procedure is based on a month by month analysis rather than a single month representing the entire summer season and another month representing the entire winter season. The thermal effect of turning light off, permitting solar radiation in the space, and conductance through the window are calculated throughout the year in the present method. In the method suggested by Rundquist, only thermal conductance was considered in the heating season and only the other two thermal factors were considered in the cooling season. An automatic light control system is assumed to be available in the space .

Energy saving in a passive solar building can be realized when comparing the energy consumption of that building to the energy consumption of a reference building. The reference building is assumed to have the same design as the passive solar building except that the passive solar window system and its wall are replaced by a neutral wall. A neutral wall is a wall that does not allow any form of heat transfer through it (an adiabatic wall).

A set of prespecified data is assumed to be available before proceeding with the energy savings calculations. This set includes:

i) The level of illuminance required at the working plane, (E_w) in klux.

ii) The area represented by each reference point, (A_t) in m^2.

iii) The coefficient of utilization of the luminaire used, *(CU)* .

iv) The light loss factor for the luminaire, *(LLF)* .

v) The efficiency of the lamps used, (ξ_L) in lumen / watt .

vi) The ratio of lighting heat goes to plenum, *(LLP)*.

vii) The efficiency of the cooling unit used, (ξ_c) .

viii) Start and end hours for using the building, (T_{sh}, T_{eh}).

ix) Number of working days for each month, (T_d) .

x) The average outdoor temperature during the working hours for each month, (T_{out}) in K.

xi) The probability of the sky to be clear, (P_{cs}), partly cloudy, (P_{ps}) and overcast, (P_{os}) .

It is assumed that the thermal mass of the direct gain passive solar building is well designed to uniformly distribute the solar heat gain to the building in a given month throughout that month. So overheating will occur only if the monthly solar gain to the building exceeds its monthly heating requirements.

The following forms of energy consumption will be considered in the calculation of both the reference building and the passive solar building: i) Energy consumed to provide auxiliary heating to the building. ii) Energy consumed to cool the building in the case of over heating. iii) Energy consumed to illuminate the building. These levels of consumptions will be reflecting the dynamic availability of daylight and solar energy in the building. These availability has been discussed in sections 3.2 through 3.4, yet their effect on energy consumption is to be discussed. The 21^{st} day of each month is assumed to be a true representative of the average day of that month.

The effect of daylighting

The following procedure is used to calculate the electric lighting energy consumption in the presence of daylight and the resulting internal heat gain to the space.

- For the 21^{st} day of each month, the available daylight illuminance at each reference point is calculated on hourly basis for each sky condition (clear, partly cloudy, and overcast).

- The consumption in lighting under each sky condition = $(E_w * (T_{eh} - T_{sh}))$ - amount of lux-hour covered by daylight [the hatched area shown in Fig.3.25].

- The average daily electric light energy consumption (C_{ave}) =

$$(C_{cs} \,.\, P_{cs}) + (C_{ps} \,.\, P_{ps}) + (C_{os} \,.\, P_{os}) \qquad (3.50)$$

where C_{cs}, C_{ps}, and C_{os} are the lighting consumption under clear, partly cloudy, and overcast sky conditions.

- The monthly average consumption for lighting at each reference point (C_L) =

$$(T_d) \,.\, (C_{ave}) \,.\, (A_s) \,/\, ((CU) \,.\, (LLF) \,.\, (\xi_L)) \qquad (3.51)$$

Where C_L units is [Watt.hr] .

- The monthly internal heat gain to the space due to lighting (q_L) can be calculated from the following equation :

$$(q_L) = (C_L) \,.\, (LLP) \,.\, (1 - (\xi_L / 683)) \quad \text{[Watt.hr]} \qquad (3.52)$$

where 683 is the maximum lumens that can be given per watt.

Both energy consumption for lighting and the consequent internal heat gain will reach their maximum value in the case of the reference building as daylight does not exist in the building.

The effect of the solar energy

The heat gain from solar energy reduces the heating requirement of a building during the heating months but it increases the cooling requirement during the cooling months. The monthly total solar energy received by a given building, **q_s** can be calculated as

$$q_S = N_d \,.\, Q_d \,.\, K_T \quad \text{[Watt.hr]} \qquad (3.53)$$

Where Q_d = the available solar energy in the room from its windows during the day, and it can be determined by summing up the hourly solar radiation, Q (Eq. 3.48) for the whole day.

K_T = the monthly average clearness index.

N_d = the number of day in the month.

Knowing the effect of both daylight and solar energy, the actual building energy consumption can be calculated for each month and then summed up to get the annual energy consumption. The following procedure is used to determine the monthly consumption.

- Calculating the building monthly heat loss from the envelope, q_e using the equation

$$\begin{aligned} q_e = \; & (A.U)_T \,.\, DD \,.\, 24 && \{\text{Envelope conductance}\} \\ & + (F2 \,.\, P \,.\, DD \,.\, 24) && \{\text{Floor conductance}\} \\ & + (1200 \,.\, v \,.\, DD \,.\, 24) && \{\text{Infiltration [sensible]}\} \\ & + (2808 \,.\, v \,.\, \Delta w \,.\, 30 \,.\, 24) && \{\text{Infiltration [latent]}\} \\ & [\text{Watt.hr}] && \end{aligned} \qquad (3.54)$$

Where $(A.U)_T$ = the sum of the area of each building envelope components multiplied by its conductance, W/°k

$F2$ = the floor heat loss coefficient W/m.°k per m length of perimeter.

P = the perimeter of exposed edge of floor, m.

v = the volume of outdoor air entering the building, L/s.

Δw = the humidity ratio difference.

- Both the internal heat gain from lighting and the solar heat gain in a building are to be subtracted from the previous values to get the monthly auxiliary required energy for heating, $C_{aux,h}$.

$$C_{aux,h} = q_e - q_L - q_s \qquad \text{[Watt.hr]} \qquad (3.55)$$

- If the inte·: .l heat gain from lighting and solar energy has exceeded the heat loss from the building envelope, the building may then need to be cooled down. An hour by hour analysis ⁱs made using the transfer function method [49] to determine the effect of the hourly solar gain to the building, the window and wall conductance, and the internal heat gain from lighting on the energy consumed to cool down the building. The following steps were taken in the calculation process:
    - Reading the outside temperature for each hour in the day 21[st] of the month, t_o, and consequently calculating the heat gain or loss by conduction for the window during each hour in the day, $q_{n,t}$ by

    $$q_{n,t} = (A.U)_n \, (t_{o,t} - t_i) \qquad \text{[Watt]} \qquad (3.56)$$

    Where $t_{o,t}$ = the hourly outdoor air temperature.

    t_i = the constant room air temperature.

    $A.U$ = the window area multiplied by its conductance.

    - Assessing the average solar energy incident on the window's wall surface, G_t in each hour where

    $$G_t = I_t \; . \; K_T \qquad \text{[Watt/m}^2\text{]} \qquad (3.57)$$

    Where I_t can be determined using the equation (3.36).

    K_T is the monthly average clearness index.

The value of G_t can be utilized in the following equation to calculate the hourly sol-air temperature, t_e [54].

$$t_e = t_o + \beta \frac{G_t}{h_o} + \varepsilon \frac{\Delta R}{h_o} \qquad (3.58)$$

Where h_o is the outside surface coefficient of heat transfer.

β is the absorptance of the wall surface.

ε is the surface emittance of the wall.

ΔR is the difference between the long wavelength radiation incident on the surface from the sky and the radiation emitted from a black body at outdoor temperature in Watt/m^2 .

The term $\varepsilon\Delta R$ / h_o varies from about zero for unobstructed vertical surface to about 4°c for horizontal surface while the rate β / h_o varies from about 0.026 for light coloured surface to about 0.053 m^2.°C/w for dark surfaces.

The hourly sol-air temperature is used in determining heat gain through the passive solar wall containing the window, $q_{L,t}$ by using the equation [49]

$$q_{w,t} = A_w \left[\sum_{n=0} b_n \left(t_{e,t-n\Delta} \right) - \sum_{n=1} d_n \left(\frac{q_{L,t-n\Delta}}{A} \right) - t_i \sum_{n=0} C_n \right] \qquad (3.59)$$

Where A_w is the area of the solar collecting wall.

t_i is the room inside temperature.

b_n ,d_n ,C_n are shown in Table 3.4. [54]

- Calculating at each hour the average solar heat gain to the building due to direct sun, $q_{S,t}$ where

$$q_{S,t} = Q \cdot K_T \quad \text{[Watt]} \qquad (3.60)$$

(Get Q from eq. 3.48)

- Calculating the hourly internal heat gain due to lighting, $q_{L,t}$ by

$$q_{L,t} = \Sigma_{1\rightarrow m} (E_w - E_r)\,(LLP)\,(1-(\xi_L\,/683))\,/\,((CU)\,(LLF)\,(\xi_L\,)) \quad \text{[Watt]} \qquad (3.61)$$

where m is the number of the reference points and

E_r is the available daylight illuminance on the reference point.

- The hourly consumption for cooling the building, $q_{c,t}$ can be determined with the equation

$$q_{c,t} = \Sigma_{i=1\rightarrow 4} (\, v_0\, q_{i,t} + v_1\, q_{i,t-\Delta} + v_2\, q_{i,t-2\Delta}\,) - w_1\, q_{c,t-\Delta} \quad \text{[Watt]} \qquad (3.62)$$

Where 1 = n, 2 = w, 3 = s, 4 = L.

Table 3.5 and Table 3.6 are to be used to choose the appropriate transfer function coefficient (v_0 , v_1 , v_2 , w_1) for each heat gain component.

- The monthly auxiliary required energy for cooling, $C_{aux,c}$ can be determined by

$$C_{aux,c} = (N_d * \Sigma_{t=1\rightarrow 24}\, q_{c,t})\,/\,\xi_c \quad \text{[Watt.hr]} \qquad (3.63)$$

The actual building total energy consumption in any month, $C_{t,a}$ can be now calculated as:

$$C_{t,a} = C_{aux} + C_L \quad \text{[Watt.hr]} \qquad (3.64)$$

where C_{aux} can be either $C_{aux,c}$ or $C_{aux,h}$ and

C_L from equation 3.51.

Using the same procedure, the monthly energy consumption in the reference building, $C_{t,r}$ can be determined noting that the cooling load will be caused only by the internal heat gain from light. The monthly saving in the actual building energy consumption, S_{mon} can be then calculated by

$$S_{mon} = C_{t,r} - C_{t,a} \quad \text{[Watt.hr]} \tag{3.65}$$

and the annual saving by

$$S_{annual} = \Sigma_{1 \rightarrow 12} S_{mon} \tag{3.66}$$

This annual saving value is useful in the decision making during the design process as indicator for choosing among daylight and solar energy systems, and it can also be utilized in sensitivity analysis studies to determine the effect of various parameters on the overall window design.

3.6. ASSUMPTIONS IN THE MODEL

The following assumptions were taken while structuring the mathematical model:

1) No external building obstructs the daylight availability on the room fenestration.
2) The room fenestration has a rectangular shape and uniform luminance distribution.
3) The room fenestration has a narrow sill and jamb that will not reflect light inside the room.
4) The room is empty and rectangular in plan.
5) The room depth is not more than twice its ceiling height.
6) The thermal mass inside the room is designed to efficiently utilize all the available solar energy in the room.
7) The room temperature is constant throughout each month.

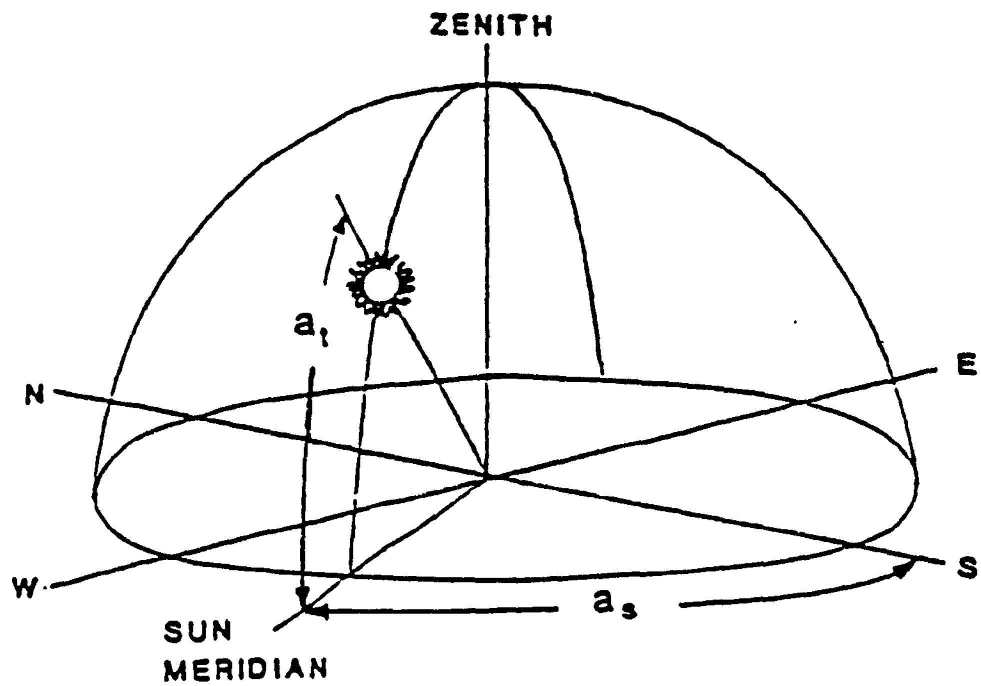


Figure 3.1 Solar altitude and azimuth (Northern Hemisphere).

TABLE 3.1 Julian Day Calculation.

Month	Days in month	J
January	31	i
February	28	31 + i
March	31	59 + i
April	30	90 + i
May	31	120 + i
June	30	151 + i
July	31	181 + i
August	31	212 + i
September	30	243 + i
October	31	273 + i
November	30	304 + i
December	31	334 + i

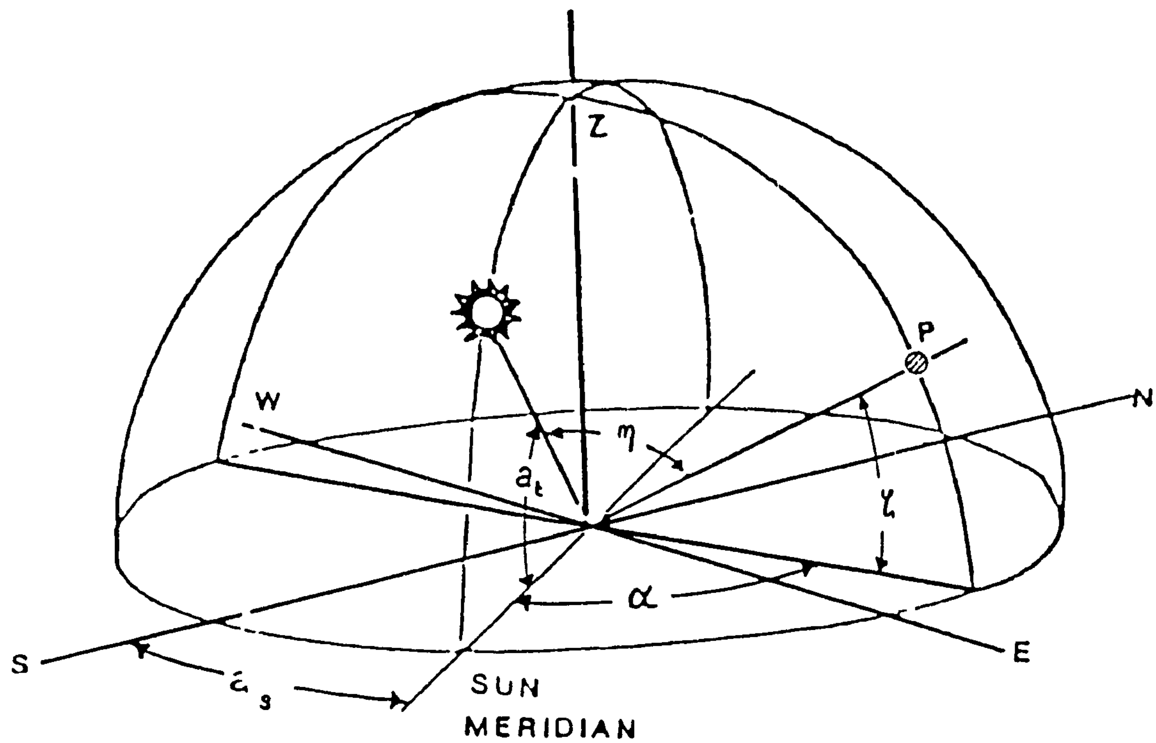


Fig 3.2 Angles Used to Determine the Sky Point Position Related to the Sun Position.

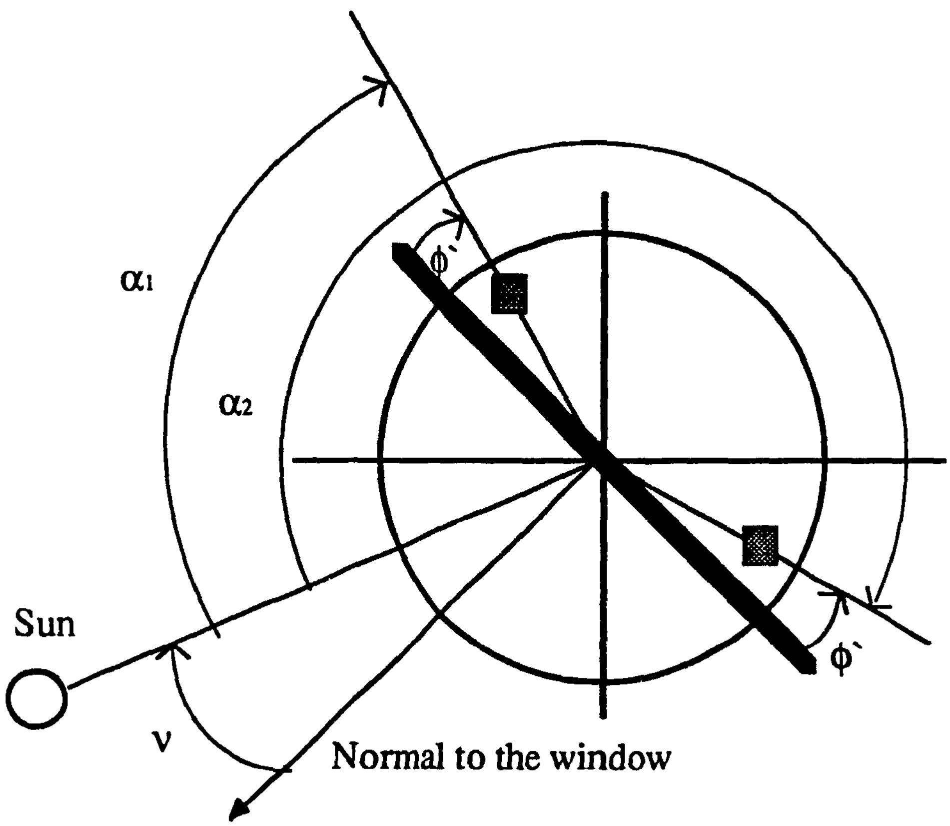


Fig 3.3. Angles used to calculate the horizontal sky illumination on tilt surface

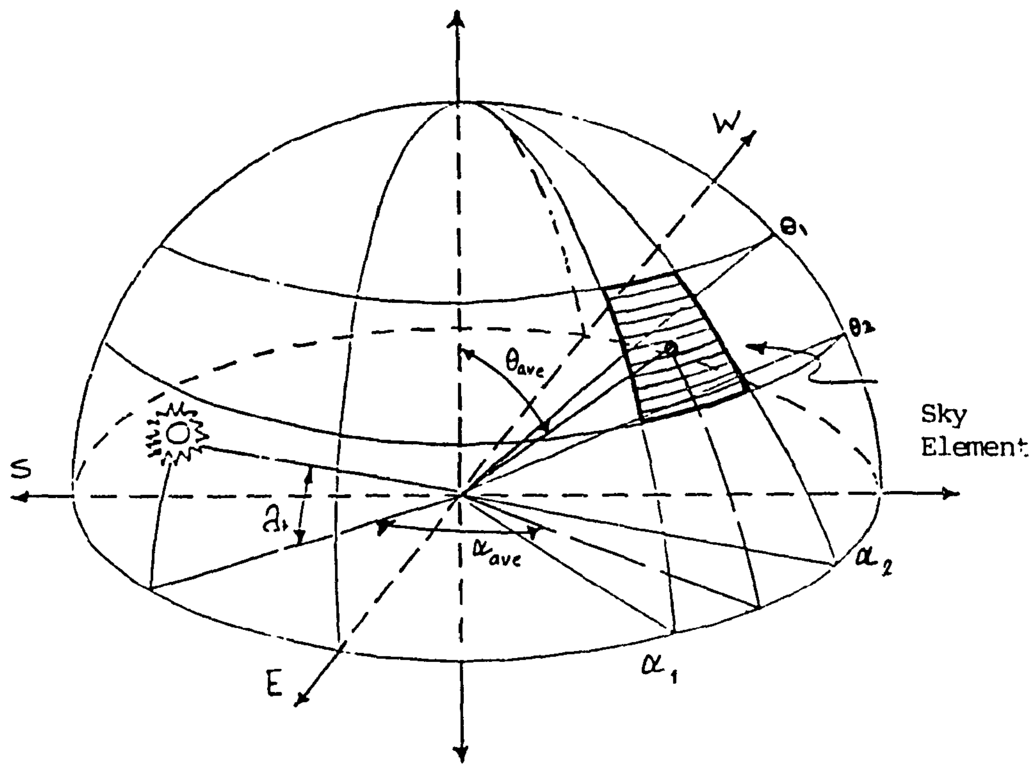


Fig 3.4 Angles Used to Calculate the Sky Illuminance on a Surface.

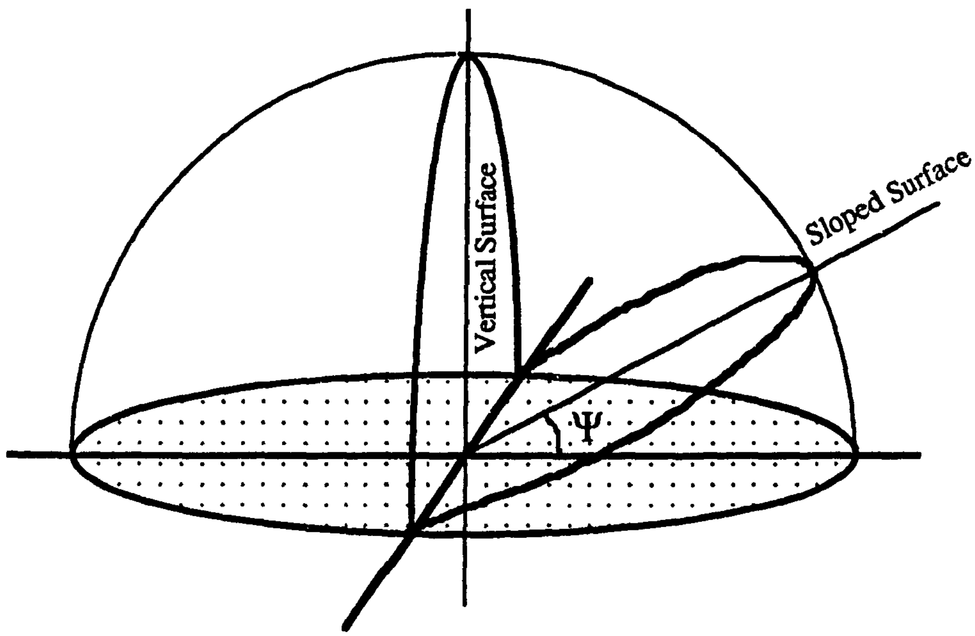


Fig 3.5 Angle between the Sloped Surface and the Horizontal Plane.

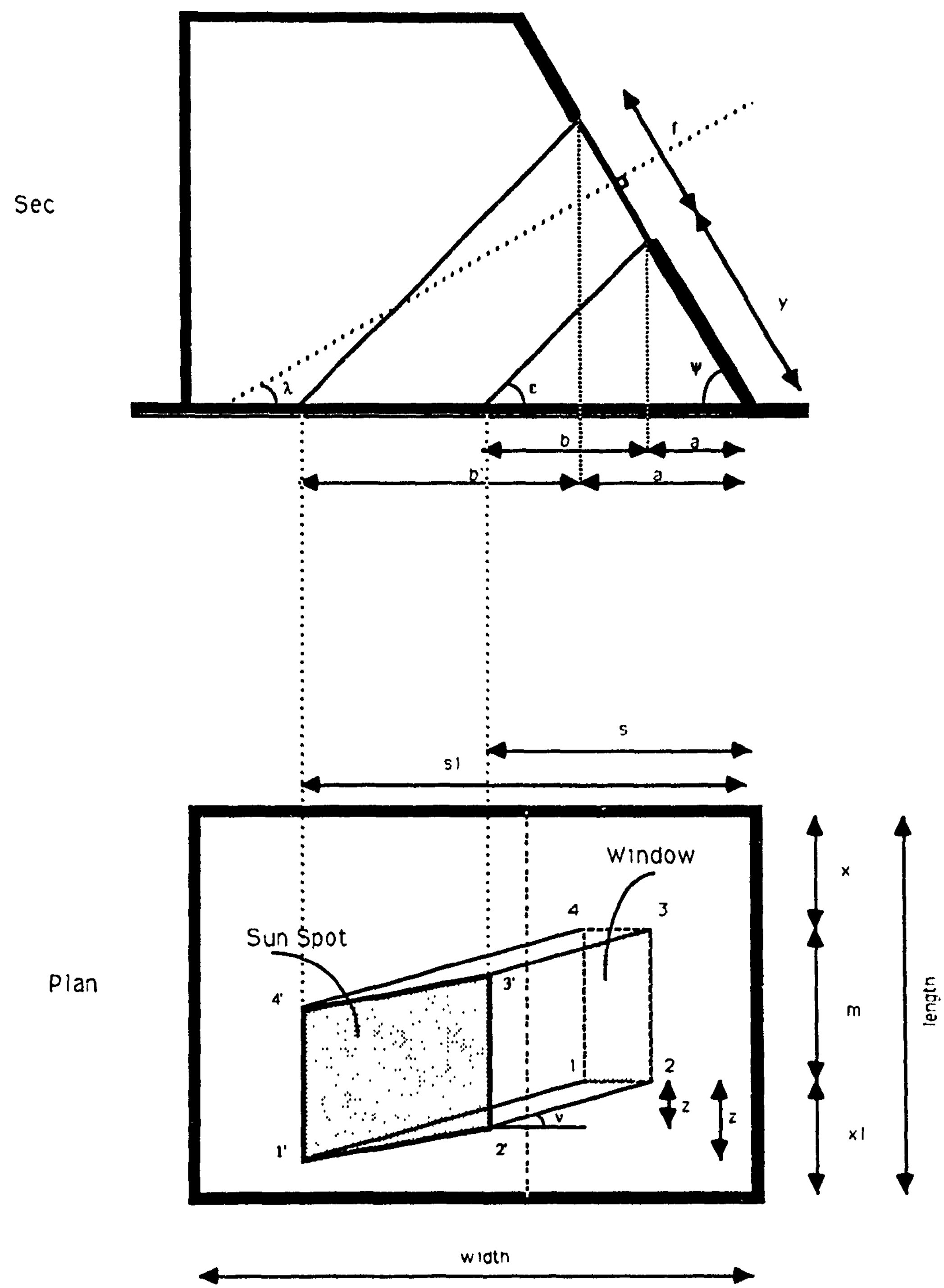


Fig 3.6 Parameters for Sun Position in the Room

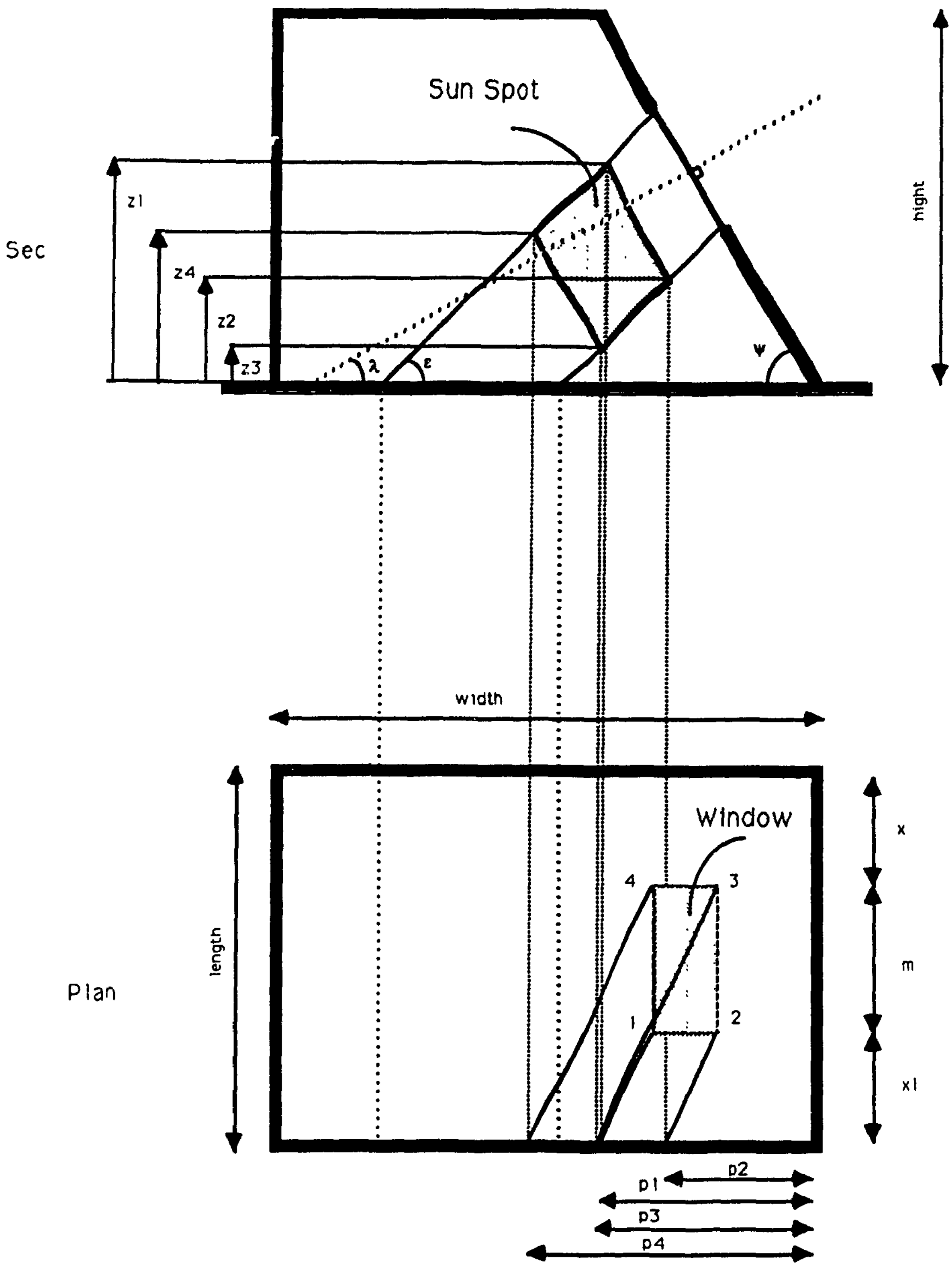


Fig 3.6 Parameters for Sun Position in the Room

(continued)

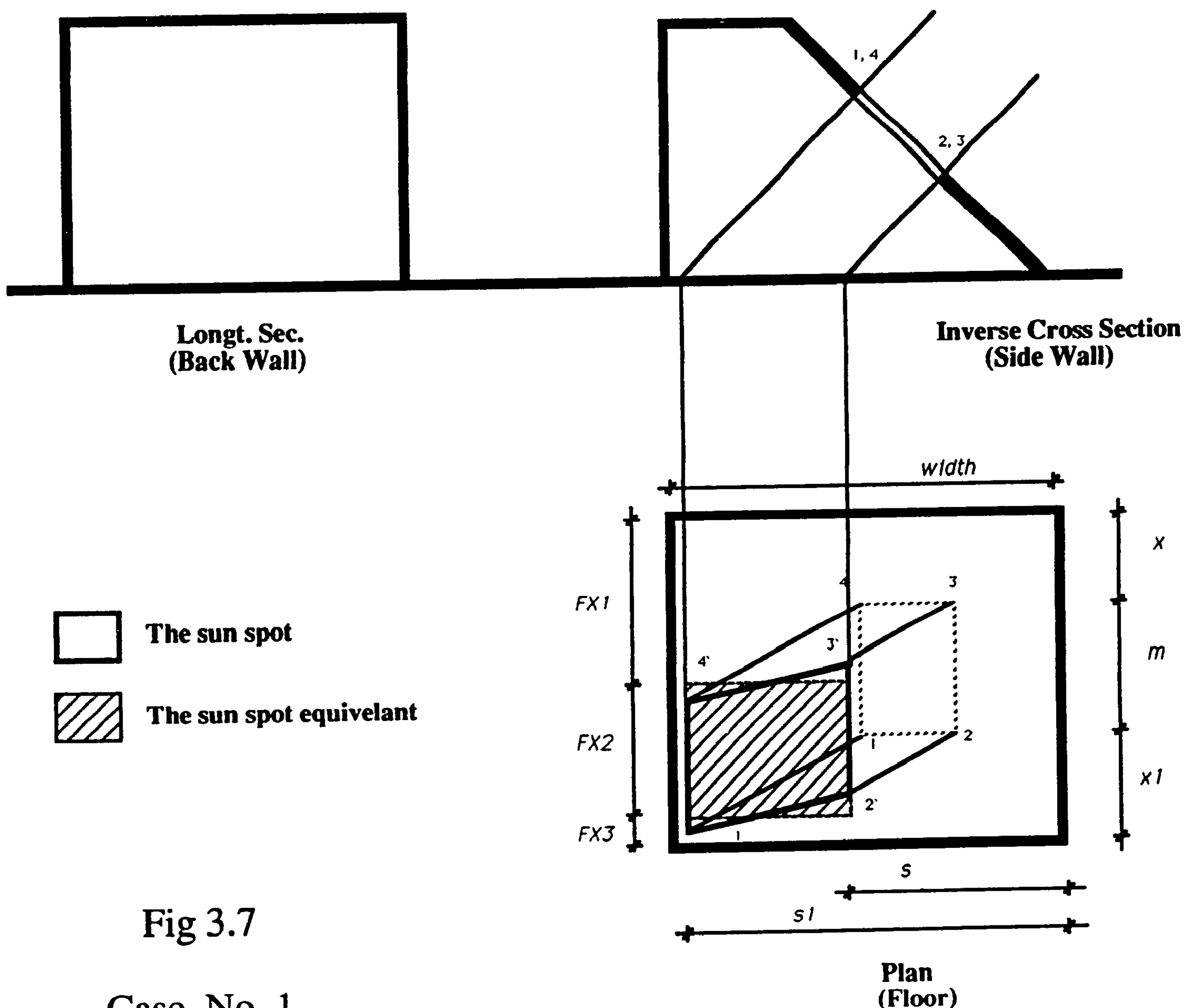


Fig 3.7

Case No. 1

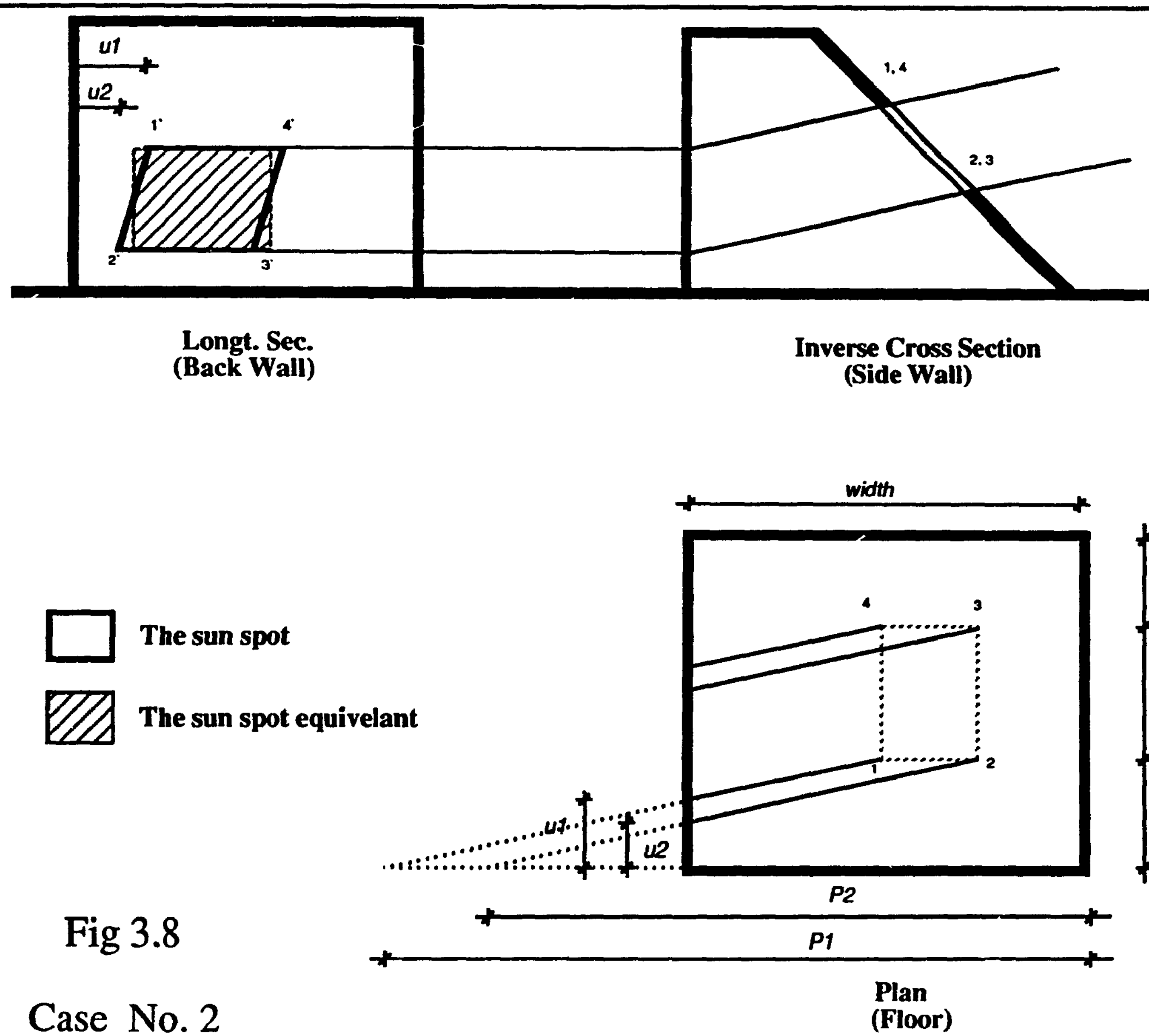


Fig 3.8

Case No. 2

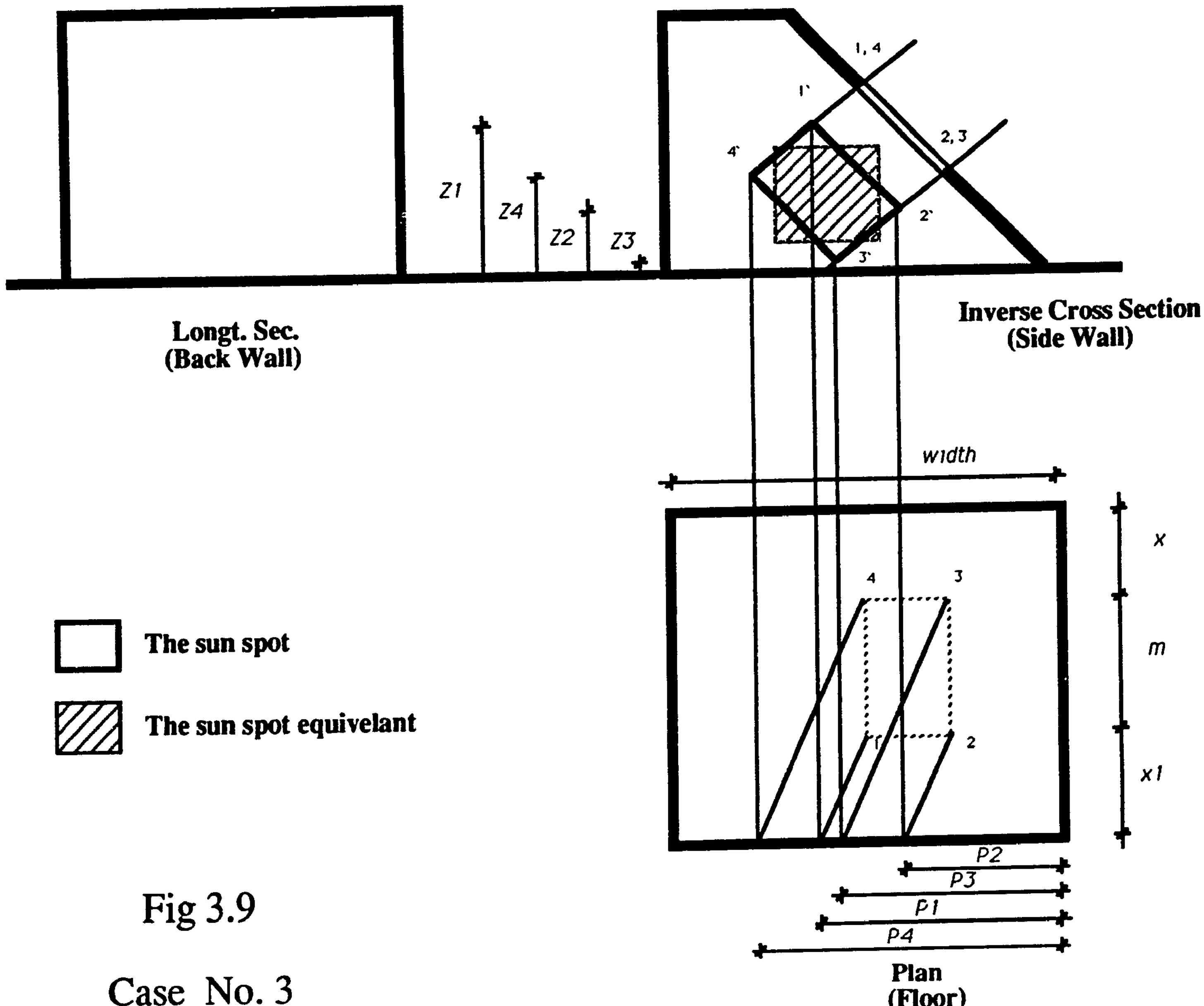


Fig 3.9

Case No. 3

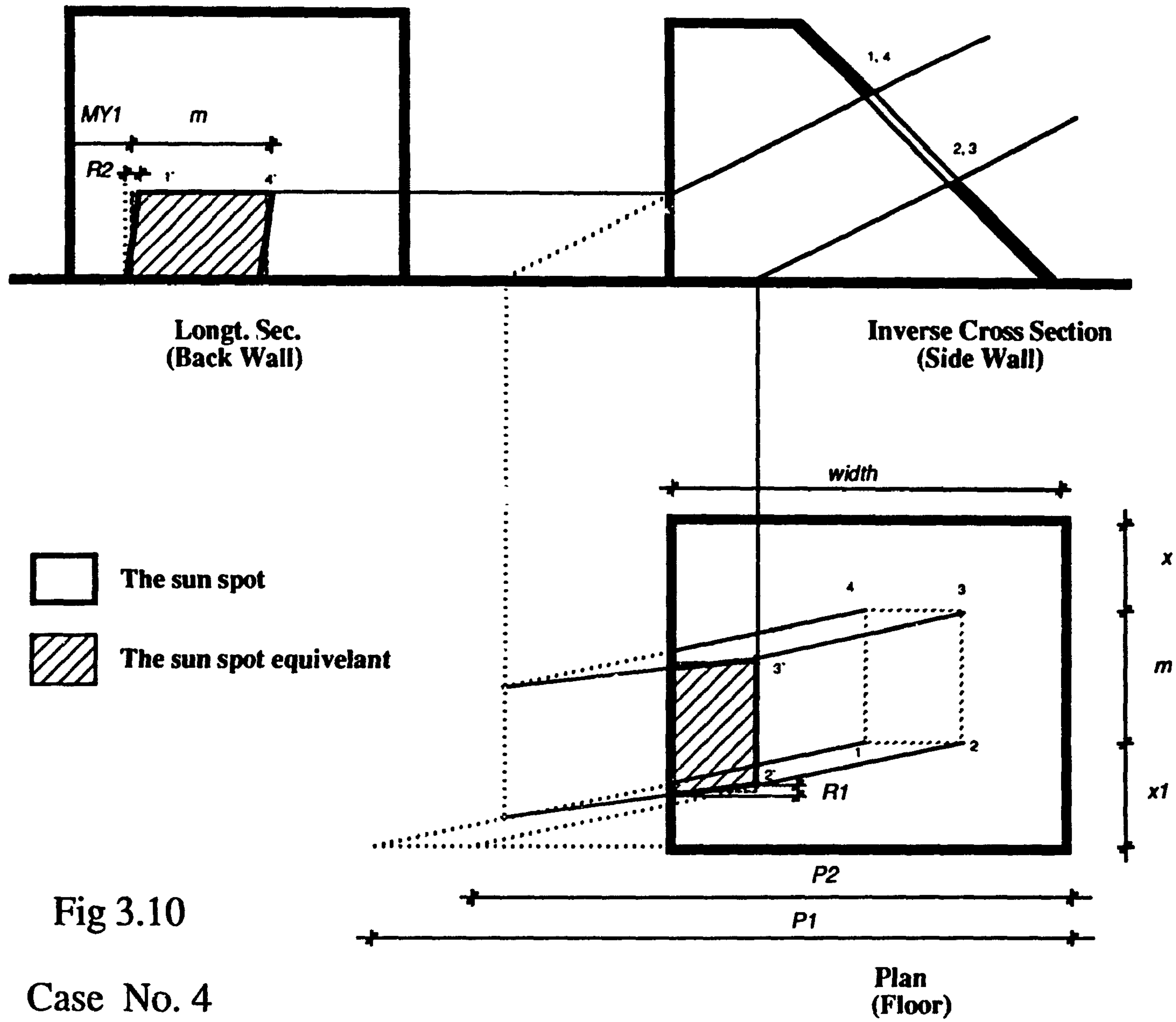


Fig 3.10

Case No. 4

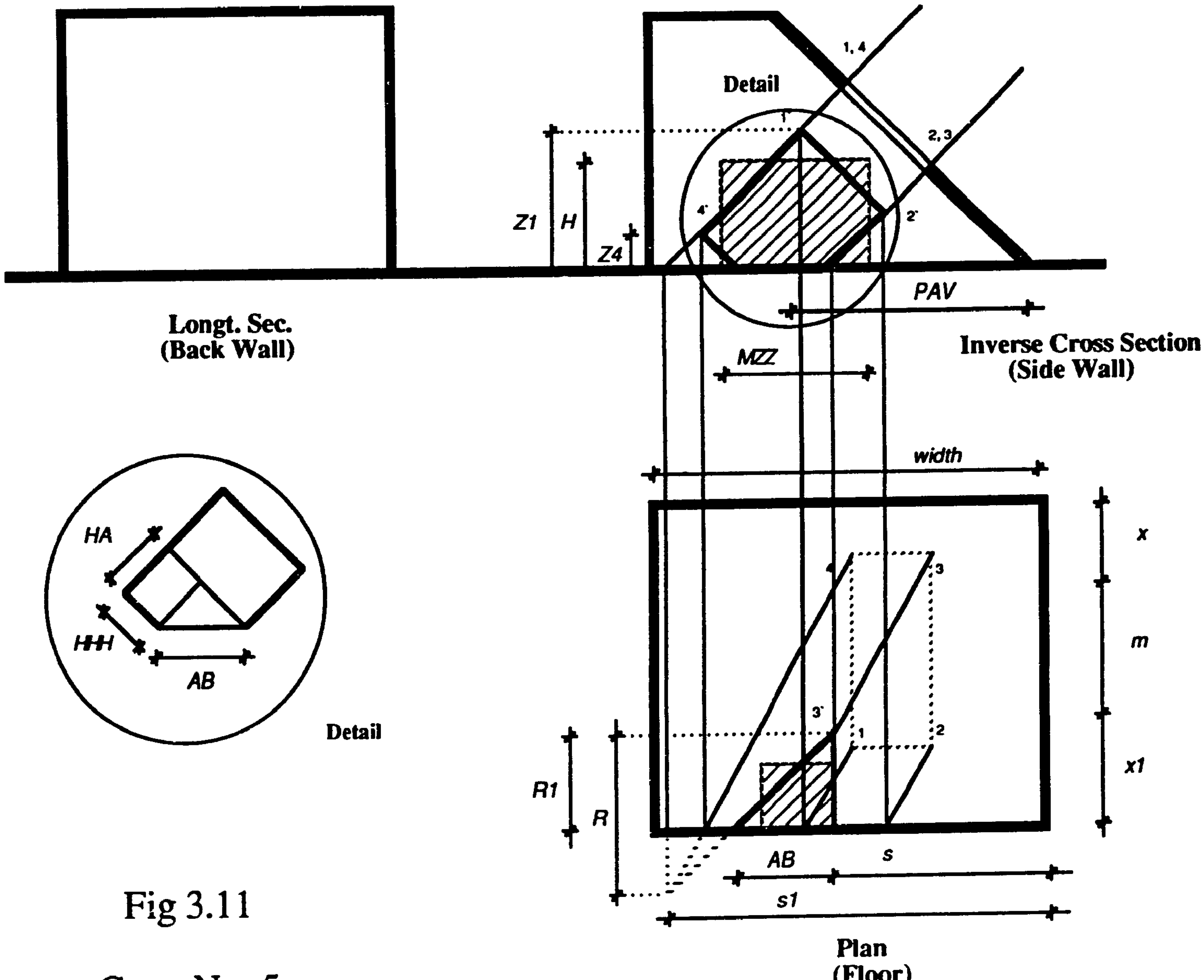


Fig 3.11

Case No. 5

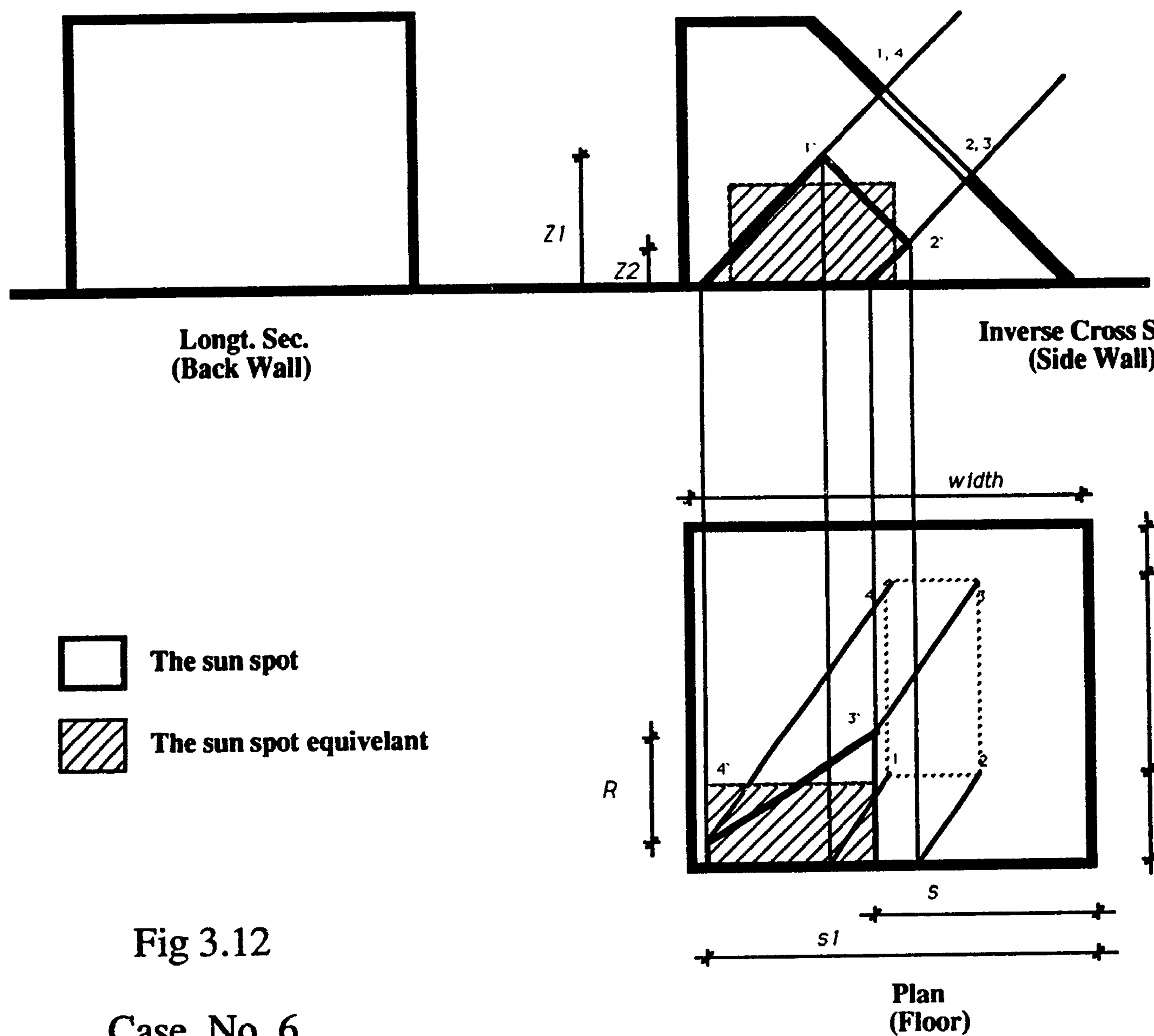


Fig 3.12

Case No. 6

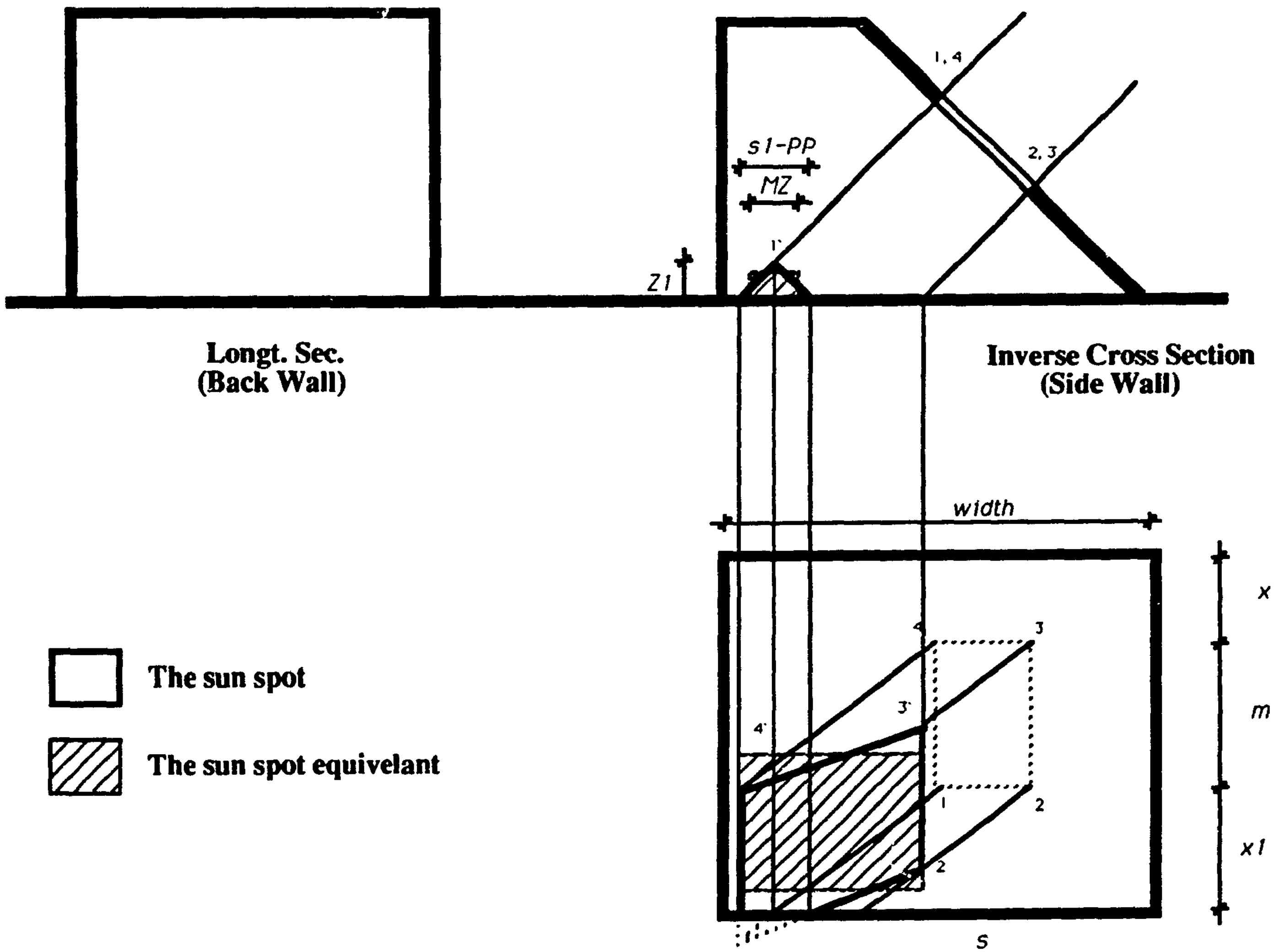


Fig 3.13

Case No. 7

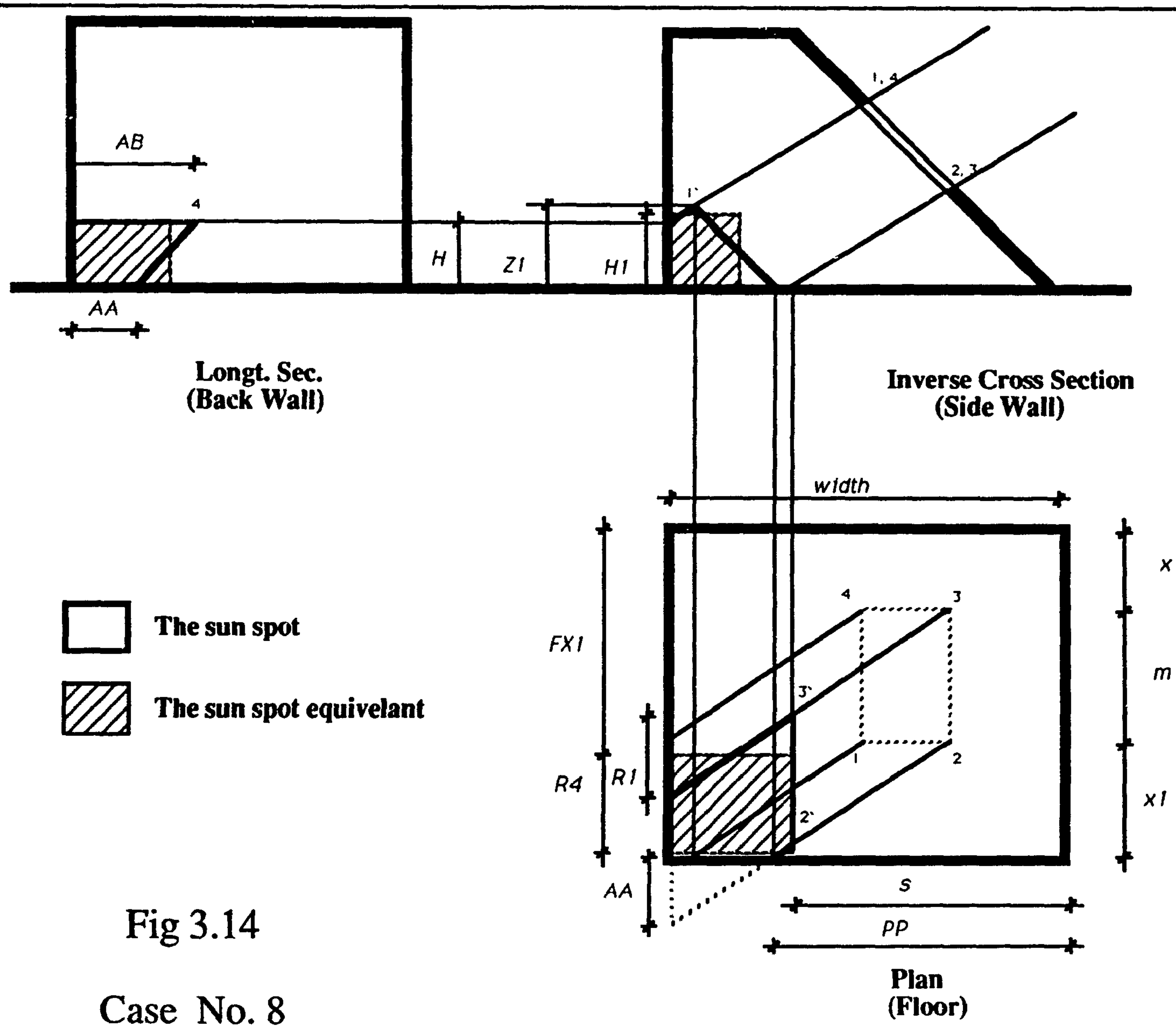


Fig 3.14

Case No. 8

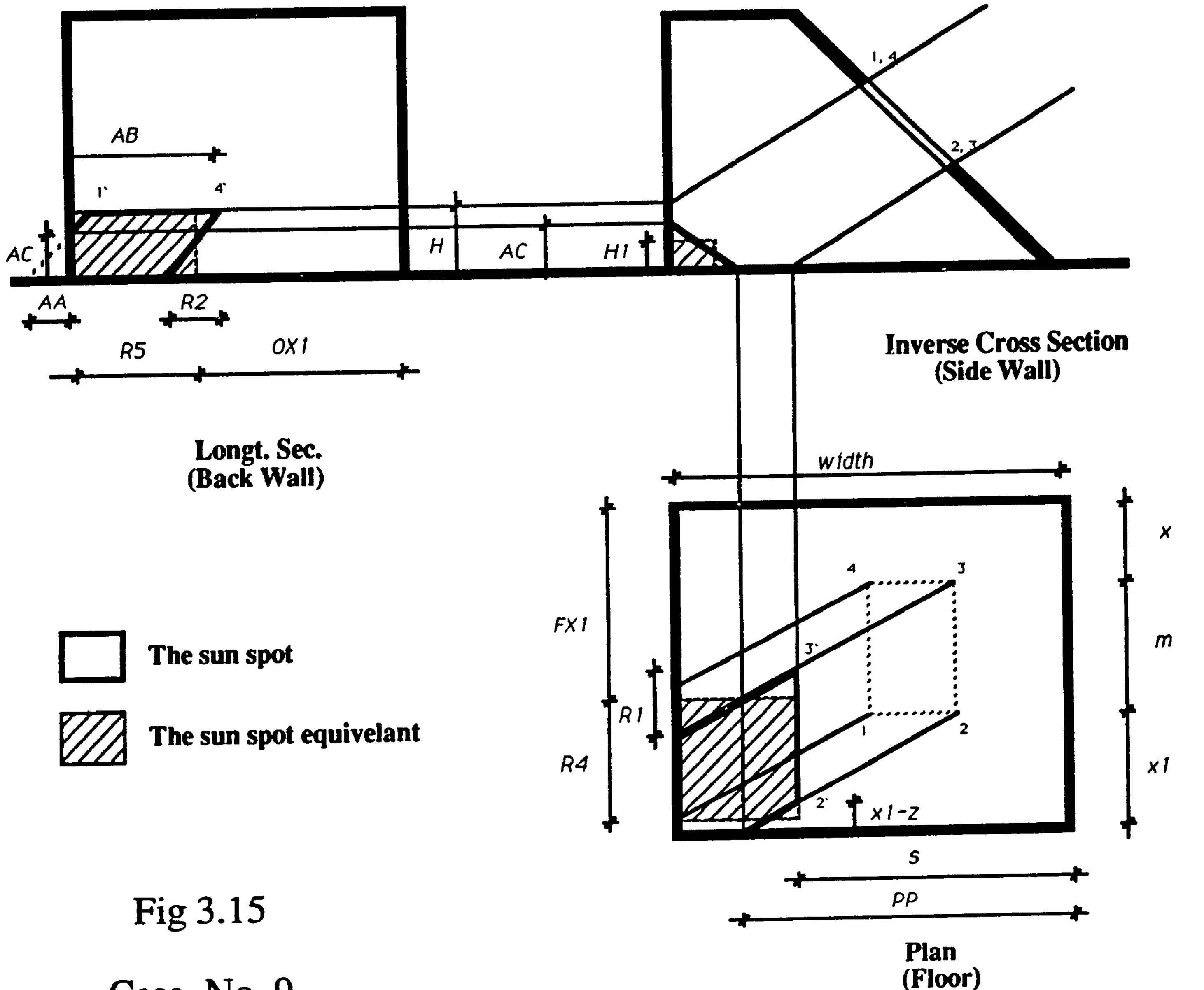


Fig 3.15

Case No. 9

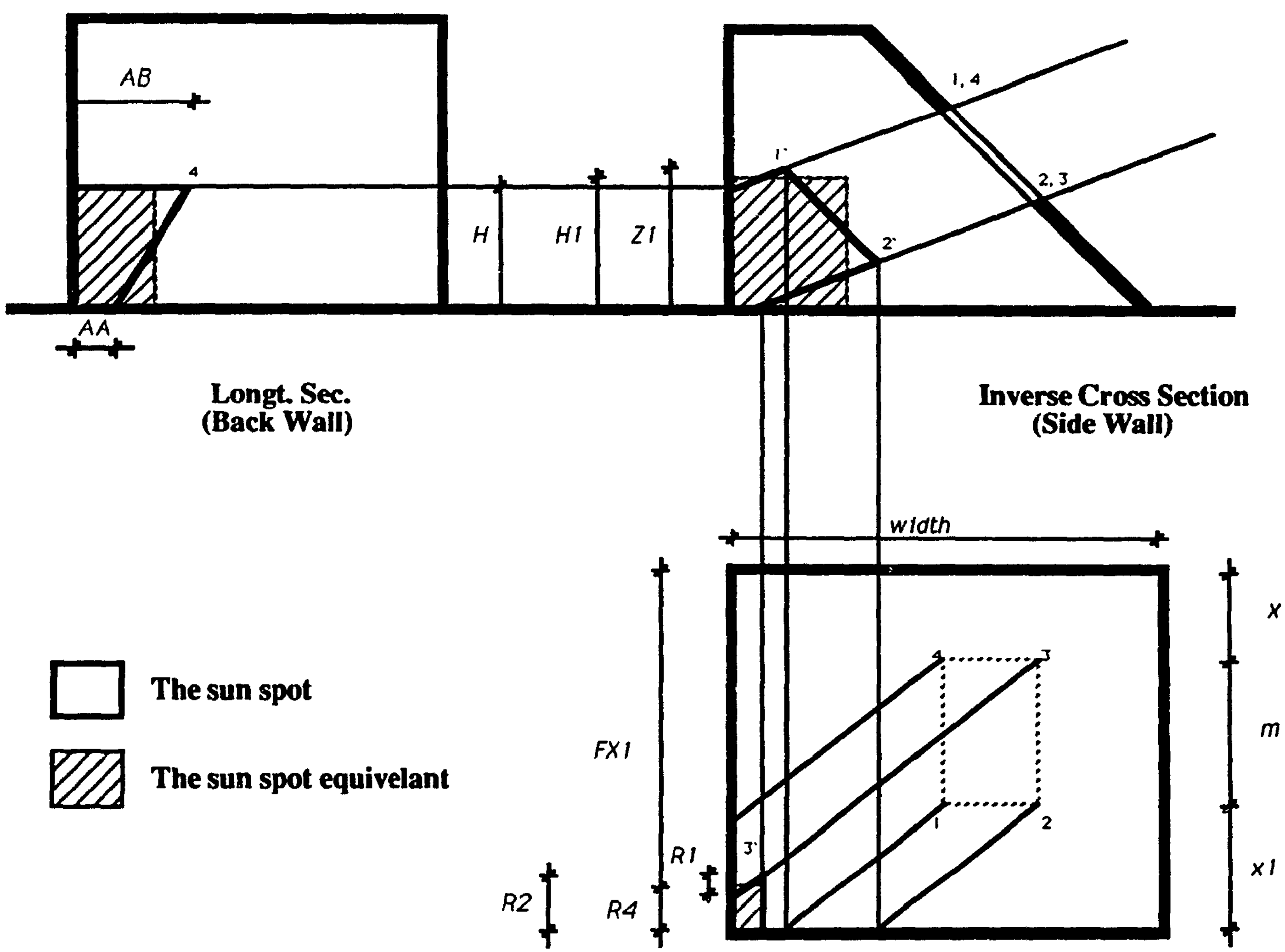


Fig 3.16

Case No. 10

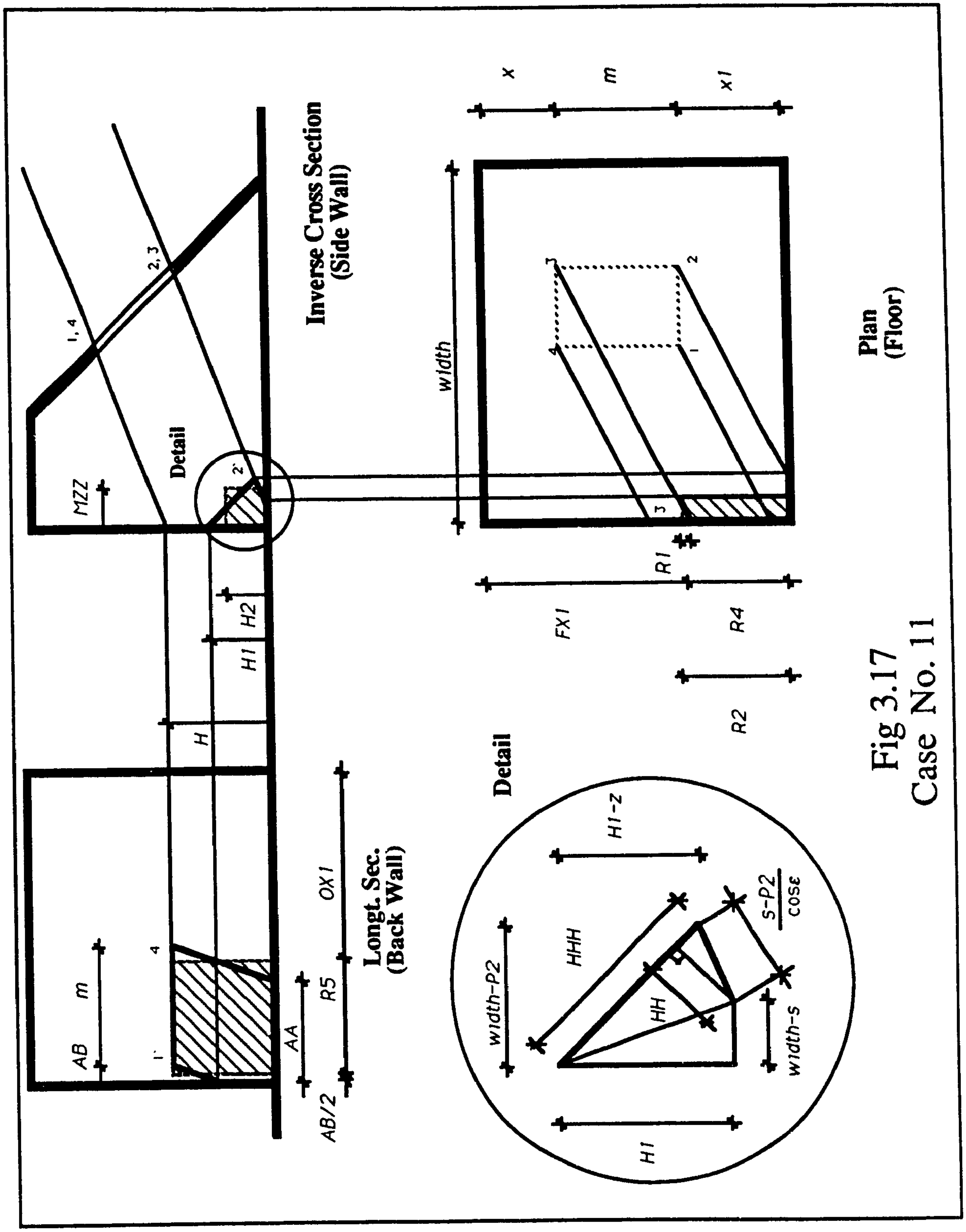


Fig 3.17
Case No. 11

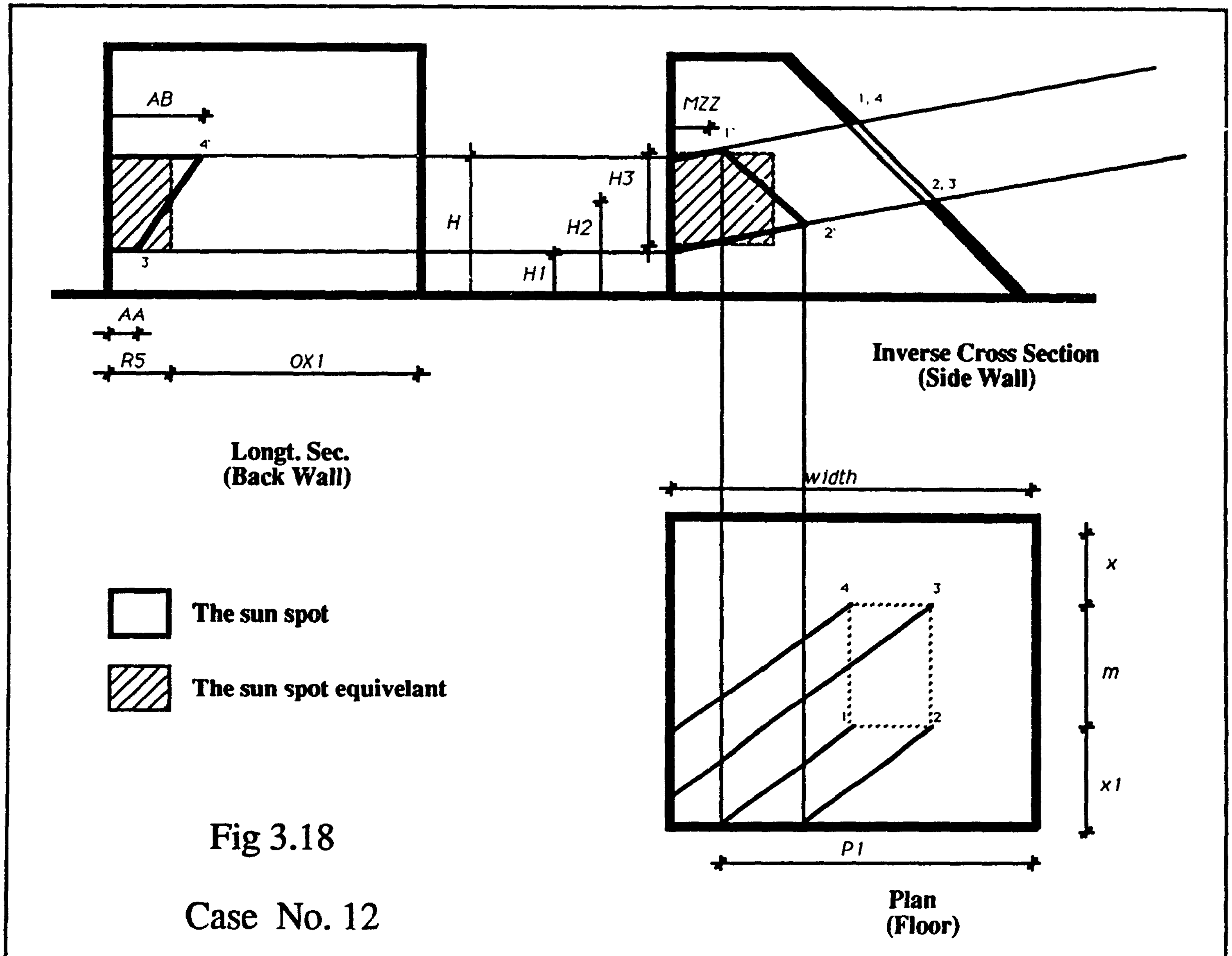


Fig 3.18

Case No. 12

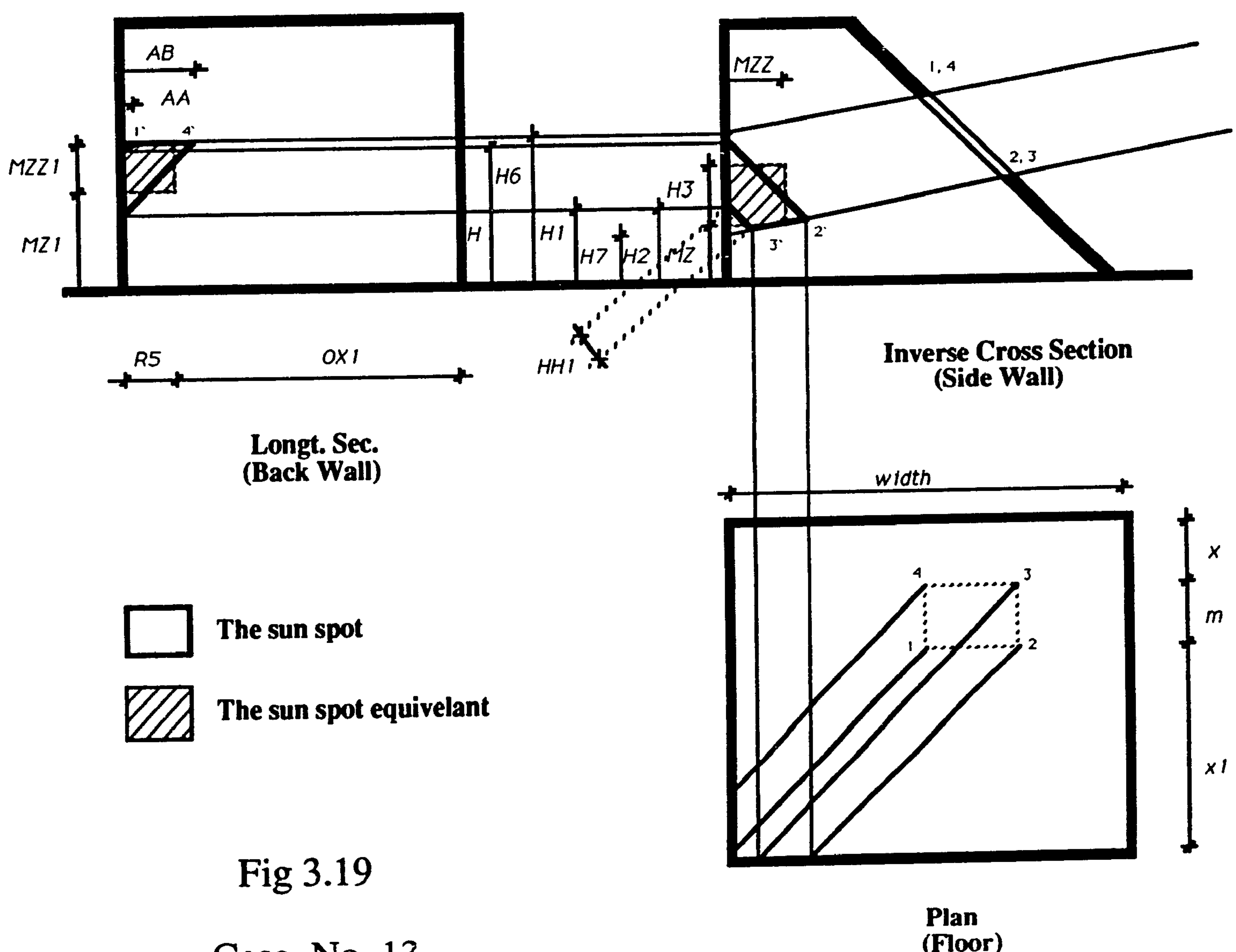


Fig 3.19

Case No. 13

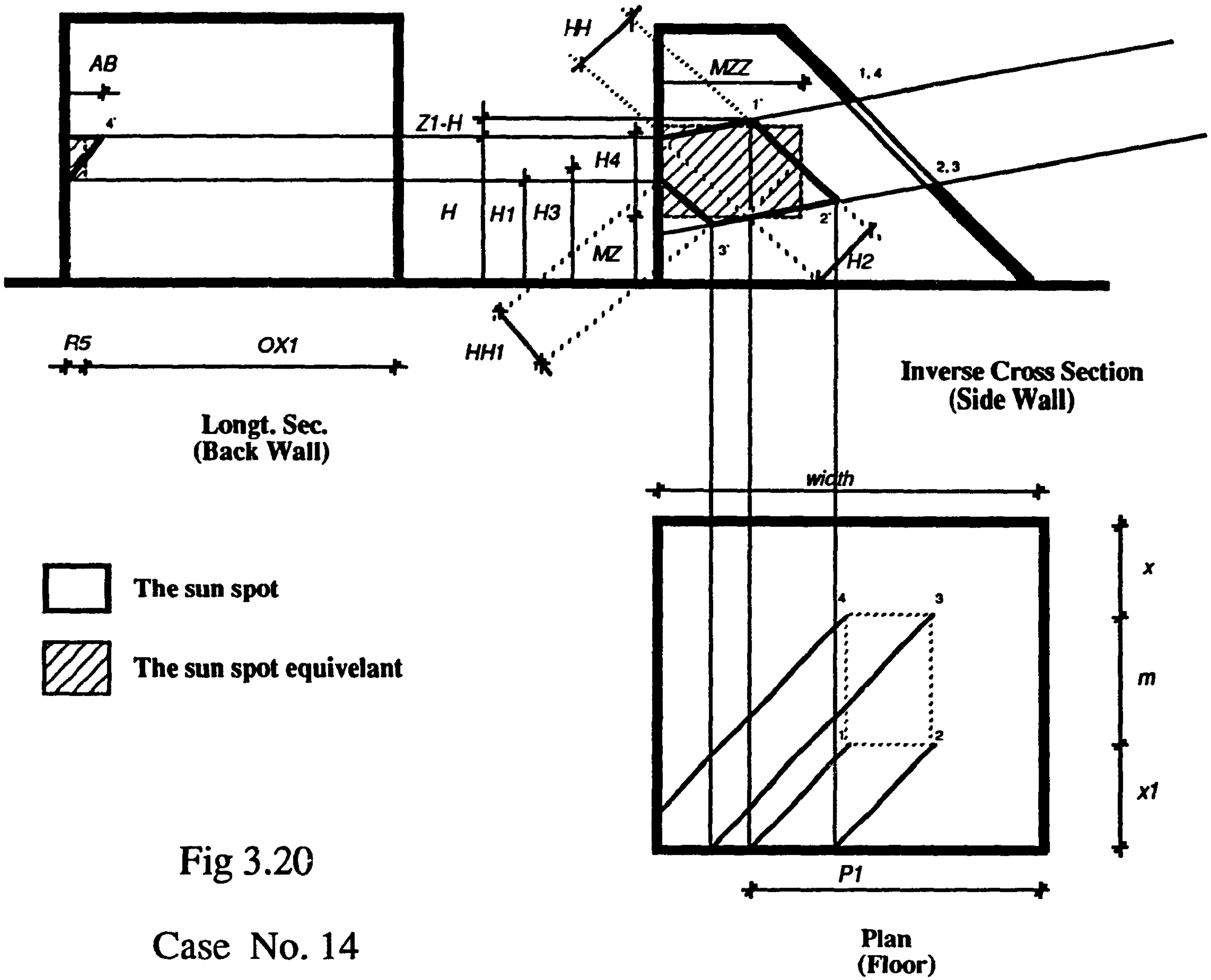


Fig 3.20

Case No. 14

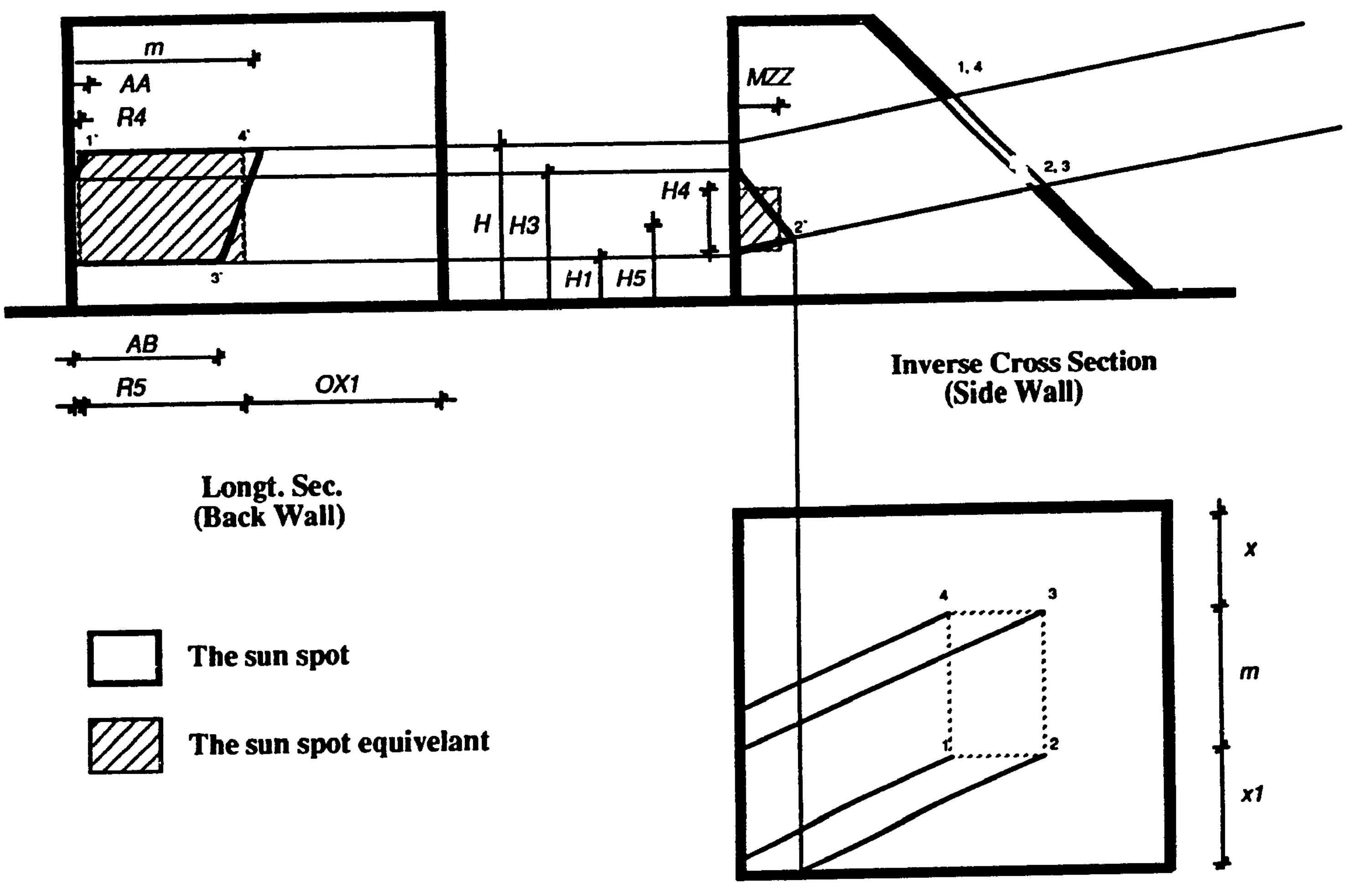


Fig 3.21

Case No. 15

Table 3.2 Possible Position of Sun Spot Corner Points

POINT POSITION / CASE #	ON FLOOR	ON PERPINDECULAR WALL	ON OPPOSITE WALL
CASE # 1	1' 4' 2' 3'		
CASE # 2			1' 4' 2' 3'
CASE # 3		1' 4' 2' 3'	
CASE # 4	1' 4' 2' 3'		1' 4' 2' 3'
CASE # 5	1' 4' 2' 3'	1' 4' 2' 3'	
CASE # 6	1' 4' 2' 3'	1' 4' 2' 3'	
CASE # 7	1' 4' 2' 3'	1' 4' 2' 3'	
CASE # 8	1' 4' 2' 3'	1' 4' 2' 3'	1' 4' 2' 3'
CASE # 9	1' 4' 2' 3'	1' 4' 2' 3'	1' 4' 2' 3'
CASE # 10	1' 4' 2' 3'	1' 4' 2' 3'	1' 4' 2' 3'
CASE # 11	1' 4' 2' 3'	1' 4' 2' 3'	1' 4' 2' 3'
CASE # 12		1' 4' 2' 3'	1' 4' 2' 3'
CASE # 13		1' 4' 2' 3'	1' 4' 2' 3'
CASE # 14		1' 4' 2' 3'	1' 4' 2' 3'
CASE # 15		1' 4' 2' 3'	1' 4' 2' 3'

A shaded corner number in a surface column indicates that this corner is on that surface

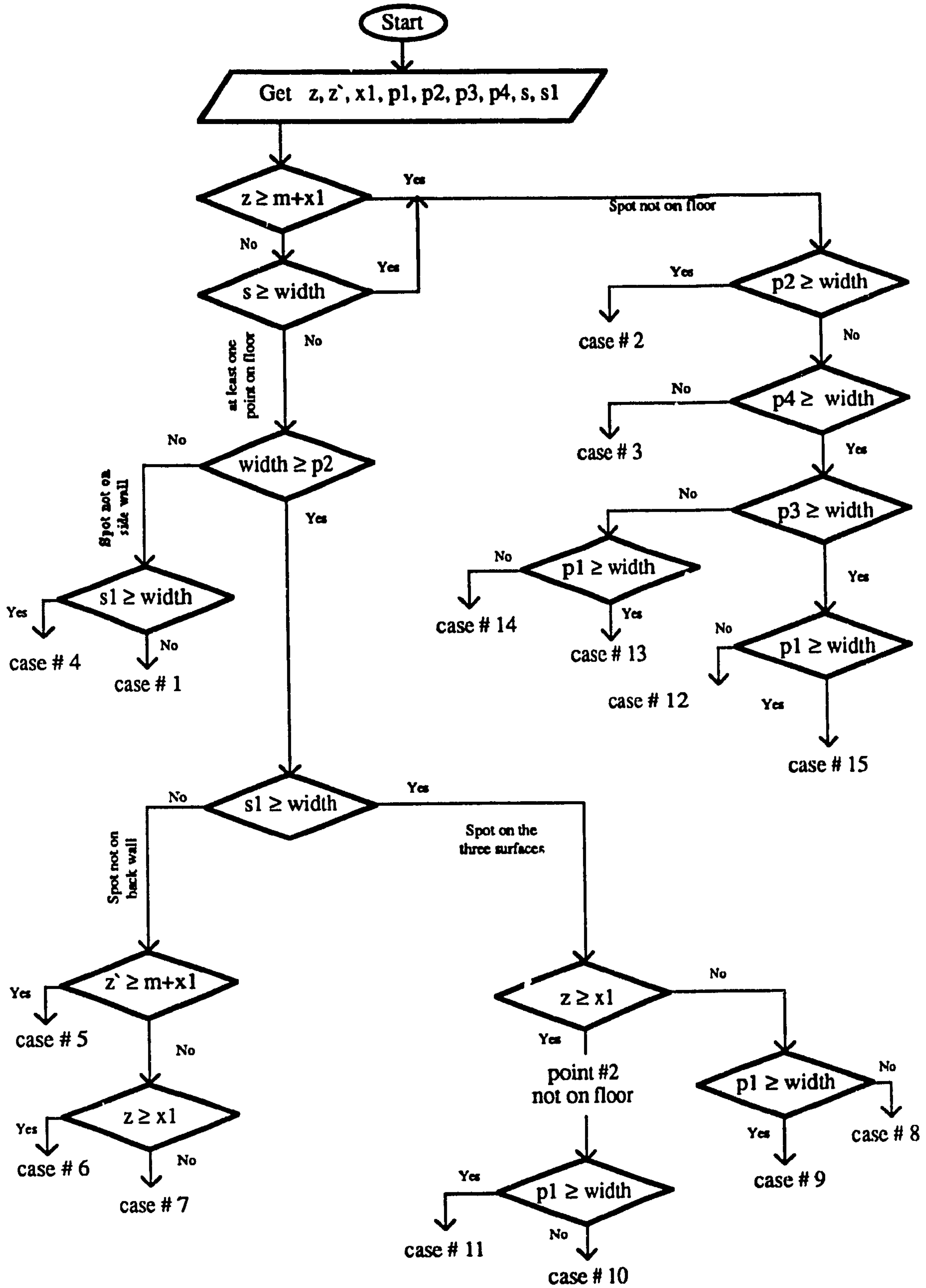
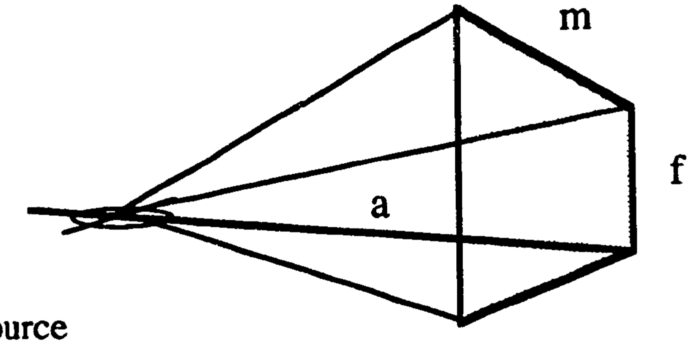
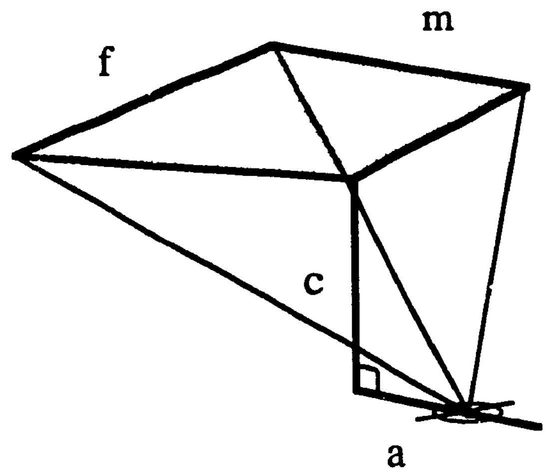


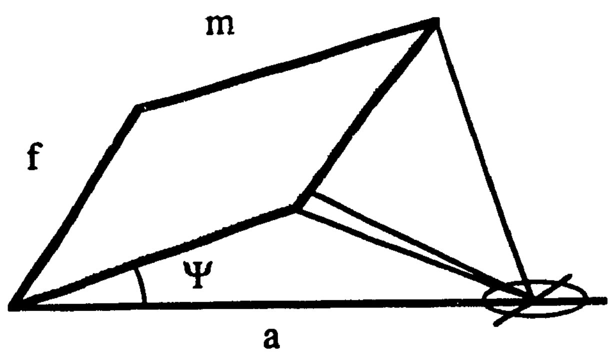
Fig 3.22 Flow Chart for Determining the Sun Spot Position Case



Perpendicular Source



Parallel Source



Inclined Source

Fig 3.23 Parameters for the Equations (3.31), (3,32), (3,33)

Table 3.3

Extraterrestrial Solar Radiation and Related Data
For the 21st day of each month
(From *ASHRAE Handbook 1981 Fundamentals*, Table 1, p. 27.2)

	I_0 W/m²	Equation of time, min.	Declination, δ, deg.	A W/m²	B Dimensionless	C Dimensionless
Jan	1396	−11.2	−20	1229	0.142	0.058
Feb	1384	−13.9	−10.8	1214	0.144	0.060
Mar	1363	−7.5	0.0	1185	0.156	0.071
Apr	1341	+1.1	+11.6	1135	0.180	0.097
May	1321	+3.3	+20.0	1103	0.196	0.121
June	1310	−1.4	+23.45	1088	0.205	0.134
July	1311	−6.2	+20.6	1084	0.207	0.136
Aug	1324	−2.4	+12.3	1107	0.201	0.122
Sept	1345	+7.5	0.0	1151	0.177	0.092
Oct	1367	+15.4	−10.5	1192	0.160	0.073
Nov	1388	+13.8	−19.8	1220	0.149	0.063
Dec	1398	+1.6	−23.45	1233	0.142	0.057

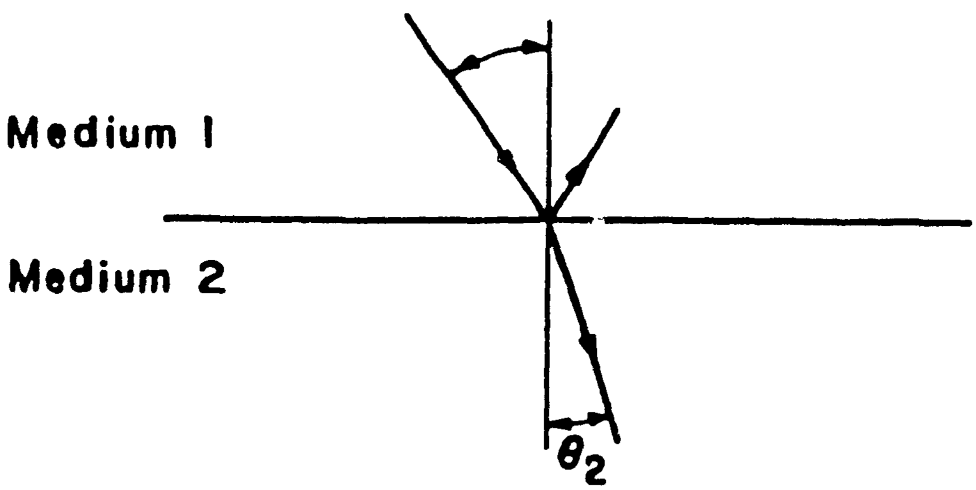


Fig 3.24 Solar Radiation Incident and Refraction Angles.

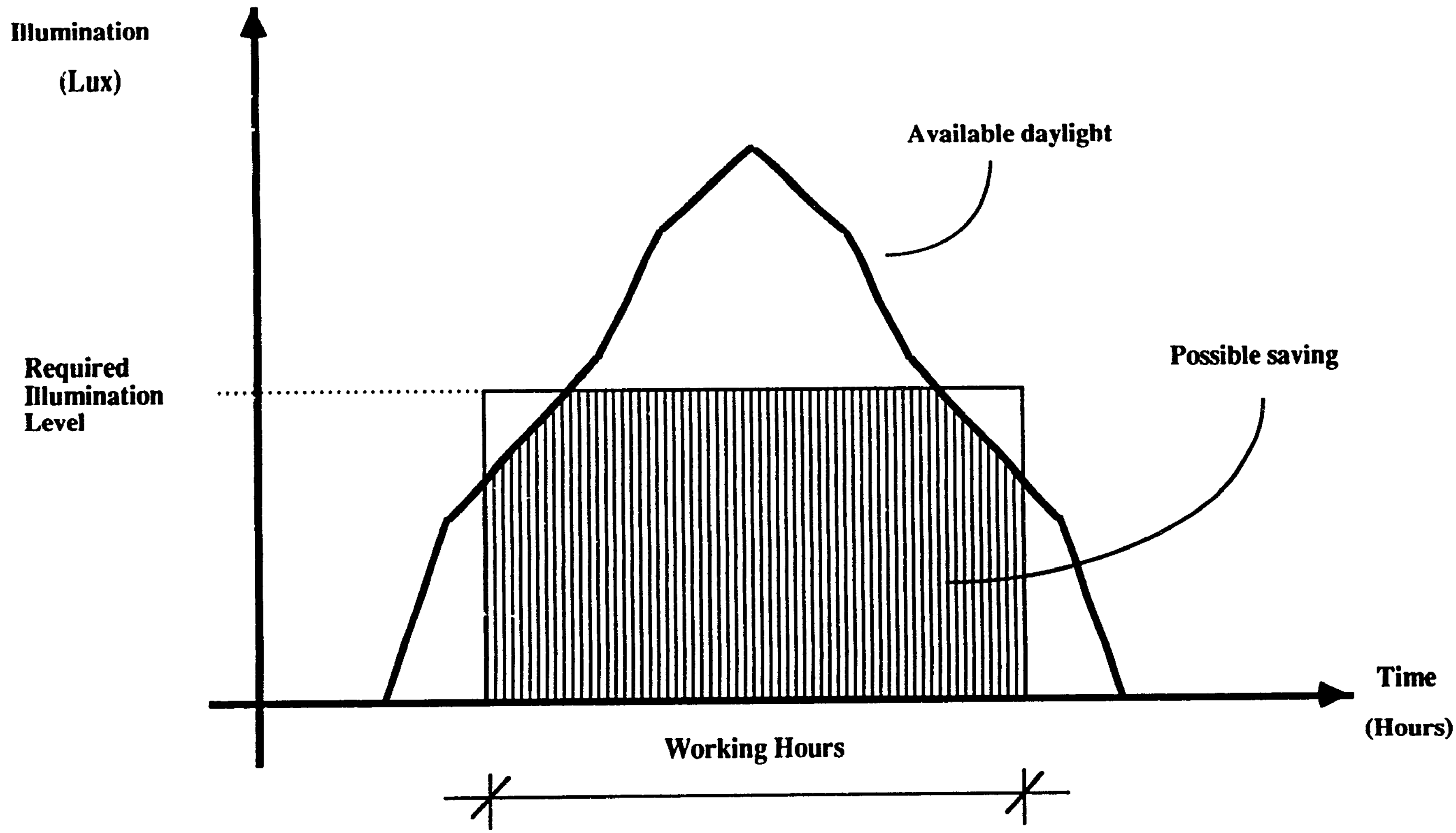


Fig 3.25 Schematic for the possible saving in lighting by available daylight.

Table 3.4 TRANSFER FUNCTION COEFFICIENTS FOR EXTERIOR WALLS (TIME INTERVAL = 1.0 hr)

Construction Description	Code Numbers of Layers		Coefficients* b_n and d_n: $n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$	$n = 6$	U	$\sum_{n=0}^{\infty} c_n$
4 in or 100 mm face brick, 2 in or 50 mm insulation, and 4 in or 100 mm lightweight concrete block	A0, A2, B3	b	0.00000	0.00046	0.00225	0.00150	0.00016			0.102	0.00437
	C2, E1, E0	d	1.00000	-1.73771	0.90936	0.11373	0.00496	0.00001			
4 in or 100 mm face brick, air space, and 4 in or 100 mm common brick	A0, A2, B1	b	0.00000	0.00086	0.00485	0.00378	0.00050	0.00001		0.301	0.01000
	C4, E1, E0	d	1.00000	-1.79201	0.98014	0.16102	0.00609	0.00003			
4 in or 100 mm face brick, air space, and 4 in or 100 mm lightweight concrete block	A0, A2, B1	b	0.00003	0.00286	0.01029	0.00504	0.00037			0.248	0.01859
	C2, E1, E0	d	1.00000	1.50943	0.65654	0.07415	0.00212				
12 in or 300 mm heavyweight concrete	A0, A1, C11	b	0.00000	0.00029	0.00303	0.00412	0.00105	0.00005		0.421	0.00854
	E1, E0	d	1.00000	-1.86853	1.09284	0.21487	0.01094	0.00009			
Frame wall with 4 in or 100 mm brick veneer	A0, A2, B6	b	0.00037	0.00823	0.00983	0.00125	0.00001			0.121	0.01969
	A6, E0	d	1.00000	-1.03045	0.20108	0.00726					
Frame wall	A0, A6, B6	b	0.01977	0.06317	0.01064	0.00006				0.124	0.09364
	A6, E0	d	1.00000	-0.25848	0.01072						
Frame wall with 3 in or 75 mm insulation	A0, A1, B1	b	0.00509	0.02644	0.00838	0.00010				0.081	0.04001
	B4, E1, E0	d	1.00000	-0.59602	0.08757	-0.00002					

Table 3.5 Room Transfer Functions: v_0 and v_1 Coefficients

Heat Gain Component	Room Envelope Construction[b]	v_0	v_1
		Dimensionless	
Solar heat gain through glass[c] with no interior shade; radiant heat from equipment and people	Light	0.224	$1 - w_1 - v_0$
	Medium	0.197	$1 - w_1 - v_0$
	Heavy	0.187	$1 - w_1 - v_0$
Conduction heat gain through exterior walls, roofs, partitions, doors, windows w/ blinds or drapes	Light	0.703	$1 - w_1 - v_0$
	Medium	0.681	$1 - w_1 - v_0$
	Heavy	0.676	$1 - w_1 - v_0$
Convective heat generated by equipment and people, and from ventilation and infiltration air	Light	1.000	0.0
	Medium	1.000	0.0
	Heavy	1.000	0.0

Heat Gain From Lights[d]

Furnishings	Air Supply and Return	Type of Light Fixture	v_0	v_1
Heavyweight simple furnishings, no carpet	Low rate; supply and return below ceiling ($V \leq 25$)[e]	Recessed, not vented	0.450	$1 - w_1 - v_0$
Ordinary furnishings, no carpet	Medium to high rate, supply and return below or ceiling ($V \geq 25$)	Recessed, not vented	0.550	$1 - w_1 - v_0$
Ordinary furnishings, with or without carpet on floor	Medium to high rate, or induction unit or fan and coil, supply and return below, or through ceiling, return air plenum ($V \geq 25$)	Vented	0.650	$1 - w_1 - v_0$
Any type furniture, with or without carpet	Ducted returns through light fixtures	Vented or free-hanging in air-stream with ducted returns	0.750	$1 - w_1 - v_0$

[a]The transfer functions in this table were calculated by procedures outlined in Mitalas and Stephenson (1967) and are acceptable for cases where all heat gain energy eventually appears as cooling load. The computer program used was developed at the National Research Council of Canada, Division of Building Research.

[b]The construction designations denote the following:

Light construction: such as frame exterior wall, 50-mm concrete floor slab, approximately 150 kg of material per square metre of floor area.

Medium construction: such as 100-mm concrete exterior wall, 100-mm concrete floor slab, approximately 340 kg/m² of floor area.

Heavy construction: such as 150 mm concrete exterior wall, 150-mm concrete floor slab, approximately 630 kg/m² of floor area.

[c]The coefficients of the transfer function that relate room cooling load to solar heat gain through glass depend on where the solar energy is absorbed. If the window is shaded by an inside blind or curtain, most of the solar energy is absorbed by the shade, and is transferred to the room by convection and long-wave radiation in about the same proportion as the heat gain through walls and roofs; thus the same transfer coefficients apply.

[d]If room supply air is exhausted through the space above the ceiling and lights are recessed, such air removes some heat from the lights that would otherwise have entered the room. This removed light heat is still a load on the cooling plant if the air is recirculated, even though it is not a part of the room heat gain as such. The percent of heat gain appearing in the room depends on the type of lighting fixture, its mounting, and the exhaust airflow.

[e]V is room air supply rate in L/s per square metre of floor area.

Table 3.6 Room Transfer Functions: w_1 Coefficient

Room Air Circulation and S/R[a]	Room Envelope Construction[b] 50-mm Wood Floor	75-mm Concrete Floor	150-mm Concrete Floor	200-mm Concrete Floor	300-mm Concrete Floor
Type	Specific Mass per Unit Floor Area, kg/m² 50	200	370	590	780
Low	−0.88	−0.92	−0.95	−0.97	−0.98
Medium	−0.84	−0.90	−0.94	−0.96	−0.97
High	−0.81	−0.88	−0.93	−0.95	−0.97
Very High	−0.77	−0.85	−0.92	−0.95	−0.97
	−0.73	−0.83	−0.91	−0.94	−0.96

[a] Circulation rate

Low: Minimum required to cope with cooling load from lights and occupants in interior zone. Supply through floor, wall, or ceiling diffuser. Ceiling space not used for return air, and h = 2.3 W/(m² · °C) (where h = inside surface convection coefficient used in calculation of w_1 value)

Medium: Supplied through floor, wall, or ceiling diffuser. Ceiling space not used for return air, and h = 3.4 W/(m² · °C)

High: Room air circulation induced by primary air of induction unit, or by room fan and coil unit. Ceiling space used for return air, and h = −4.5 W/(m² · °C)

Very high: Used to minimize temperature gradients in a room. Ceiling space used for return air, and h = −4.5 W/(m² · °C)

[b] Floor covered with carpet and rubber pad; for a bare floor or if covered with floor tile, take next w_1 value down the column

CHAPTER IV

CODING THE MATHEMATICAL MODEL

A computer program was developed for the model presented in chapter III. The program runs on any IBM compatible personal computer with an operating system "DOS" version 3.3 or higher, and using "Basic" as the main programming language. The program was written using the Quick BASIC program version 4.5 and was connected in its final steps with LOTUS 123 version 2.2. The interface with the user was done using SCREEN SCULPTOR version 1.2.

4.1. FUNCTION OF THE PROGRAM CODE

This program code is designed to provide the following functions: i) to estimate the energy savings potential of daylight in passive solar buildings. ii) to provide required sensitivity analysis for assessing the influence of windows on building energy consumption. iii) to provide daylight designers with a user friendly software that helps them in designing as well as comparing design alternatives. The program is able to supply the users with general unuseful data needed in the design process. Such data includes the sky and sun illuminance on horizontal, vertical and tilted surfaces, the solar incident angle on a surface, the hour angle and the sunset hour angle, the azimuth and altitude angles of the sun for a specified solar time and building location. The program would also calculates the available illuminance level on any given point. The available solar energy during a period of time and the saving in energy consumption due to passive solar energy and daylight for any given design are also available to the user.

4.2. THE PROGRAM STRUCTURE

The program consists of a number of subroutines. Each of them performs a specific task. These subroutines are clustered and chained together in several combinations so as to provide maximum flexibility to the user. Clustering and chaining the subroutines is made by using three main modules; the Control module, the Input Data module, and the calculation module (Fig. 4.1). The control module is designed to interface with the user and to receive input information that orient the program towards achieving the user requirements. The Input Data module is organised into two sections, the city weather data section and the building data section. The calculation module is the heart of the program and consist of three routines; the main/diffuse sky routine, the direct beam illuminance routine, and the solar energy routine. Some of the results are performed through the control module and the rest are presented by another peripheral module made by the "LOTUS" program in order to show and printout graphs. This modular structure allows easier development to the different parts of the program and facilitates their future expansion or modification.

4.3. THE CONTROL MODULE

This module guides the user and permits the selection of the required task from the available list menu (see appendix B). Then it sends operating orders to the proper module and receives the results to present them. Fig. (4.2) shows the different tasks made by this module.

4.4. THE INPUT DATA MODULE

4.4.1. Capabilities and limitations

This module allows inputting the location and weather data for any number of cities where the weather data are entered on monthly bases. The program can deal with one building at a time and the building can have unlimited number of rooms. Each room should have a rectangular prism shape with possibility of only one tilted wall. In case of a room with a tilted wall, the window must be located on the tilted wall. The walls of the room should have equal reflectance. The reflectance of the floor and ceiling can be different. The window azimuth angle should be the same angle as that of the wall. A window can be positioned anywhere within the wall. If there is a window on the tilted wall, it must have the same inclination angle as the wall. Each window could have a perpendicular overhang and two similar perpendicular side fins. A window can have any type and any number of glass panes as long as its transmittance and reflectance values accounts for window type and number of panes. The maximum number of windows within one room is ten. The reference points can be situated anywhere within the room with a maximum number of ten points.

4.4.2. The Structure

The input data module includes two separate parts. The first sub-module includes the city weather and location data. The second sub-module contains building data. The city location data is stored in the form of a two dimensional matrix as shown in Fig. 4.3 and it consists of monthly average probability of a clear, partly cloudy, and overcast sky. The monthly average outdoor temperature is in Centigrade, the monthly average daily horizontal radiation is in kJ/m^2, the clearness index, and the number of heating degree days. It also includes the city latitude, longitude, altitude, and the time zone standard meridian. The building data is stored in four multidimensional matrices. The first matrix (Fig. 4.4) covers the overall building data which are the number of spaces in the building, the cost of kWh in dollars, the number of working days in each month, the indoor air temperature in each month. The second matrix (Fig. 4.5) describes the building rooms, each room is described with two dimensional matrix which includes the dimensions of the room surfaces (walls, ceiling and floor), their area and reflectance values. It also includes the number of windows, and the azimuth angle of each wall. Other data such as total number of reference points in the room, number of working hours per day, outside ground reflectance, the coefficient of utilization for luminaries, their light loss factor, their lamp efficiency, the efficiency of the light control system and the cooling unit, and the conductance value of the room walls facing the outside are included in the second matrix. The third matrix (Fig. 4.6) deals with building windows description. The windows of each room are described in two dimensional matrix where every row represents a window. The matrix includes the window wall number, the window position on the wall, width and hight of the window, its altitude and azimuth angle, the properties of its glass and the dimensions of any overhang or side fins around the window. The fourth matrix (Fig. 4.7) describes the reference points. Each room is represented in two dimensional matrix that includes the reference points positions, their corresponding areas that they cover, and the required illuminance level at the reference point.

4.5. THE CALCULATION MODULE

4.5.1. Capabilities and limitations

This module is designed to simulate both daylight and solar energy that enter the room and to calculate the energy savings from daylighting. It can model time dependent clear, partly cloudy and overcast skies taking into account the solar spot position within the room. It does not consider the effect of adjacent buildings nor window blinds. This module estimates the shading effect resulting from window overhang and side fins. As luminance of a room surface may vary considerably, each surface is divided into 3 strips of equal width having luminances that were calculated separately. The available solar energy and daylight are calculated on hour by hour basis and the degree of direct beam penetration can be assessed for any given window with any tilt angle and orientation. The angular dependence of transmittance through glazing is determined for both direct and diffuse radiation. This module allows the calculation of the energy savings due to daylight and solar energy for a whole year and for any specific month in the year. It also allows the calculation of illuminance level on any given surface at any given time and location (see sec. 4.7).

4.5.2. The Structure

The calculation module consists of three routines: the main/diffuse sky illuminance routine, the direct beam illuminance routine, and the solar energy routine.

- Main / Diffuse Sky Illuminance Routine :

This routine is connected with the control module Fig. (4.1) to receive the operating orders and sends the results back to the control module. It also organizes operating the other two routines in the calculation module. The routine answers the user requests for the daylight and solar energy design information. It also determines the illuminance effect of the window on the room surfaces and on the reference point. Fig. (4.8) shows the flow chart for this routine.

- Direct Beam Routine :

In this routine, the illuminance due to the solar beam is assessed. The calculation procedures takes into consideration the position of the sun spot within the room surface. Fig. (4.9) presents the calculation sequence of this routine.

- Solar Energy Routine :

Its main function is to determine the available solar energy in a room according to its fenestration design and orientation. Fig. (4.10) shows the flow of calculation in this module.

4.6. GRAPHICAL INTERFACE USING LOTUS

In order to present graphical output from this program, the "LOTUS" was customized to translate numerical outputs into graphics. These graphics shows the illuminance level at each reference point, the available daylight and the potential energy savings due to daylight and solar energy.

4.7. APPLICATION EXAMPLE

To illustrate the program capabilities, the following application example is introduced. The graphs shown (Fig. 4.11) are a black and white version of the colourful screens originally used in the program. Each screen will be followed by comments to show the reader which module is currently on. This example will start by introducing the program then the user will choose running the program to get information for a case study in Montreal. The user then inputs the latitude and longitude of Montreal, then instructs the program to calculate both the sky illuminance on a horizontal surface and the sun position on the 20^{th} of June at 3:30 p.m. Next, the user enters all building data needed. The building name in this case study is "Test" and it is located in Montreal with only one space (room) which is 12 x 18 m^2 in plan and 8 m in hight. The room has one window in the south wall, and there will be six reference points within the room. The window dimensions are 7 x 5 m^2 and it has an over hang. Having filled all the required data, the user is going to modify the window dimensions into 10 x 6 m^2. Finally the program is executed to calculate the available daylight, solar energy and the potential energy savings.

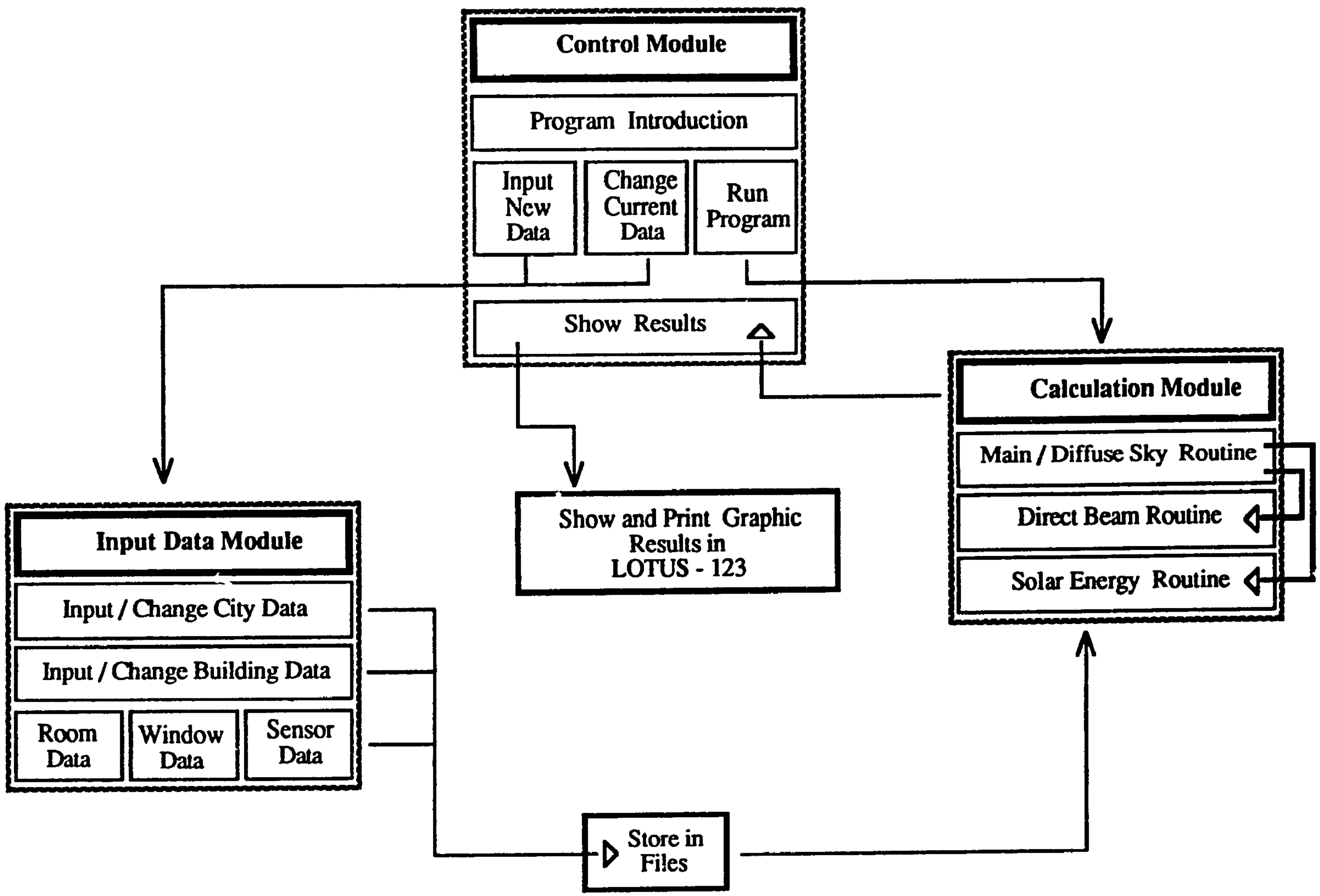


Fig 4.1 Program Structure

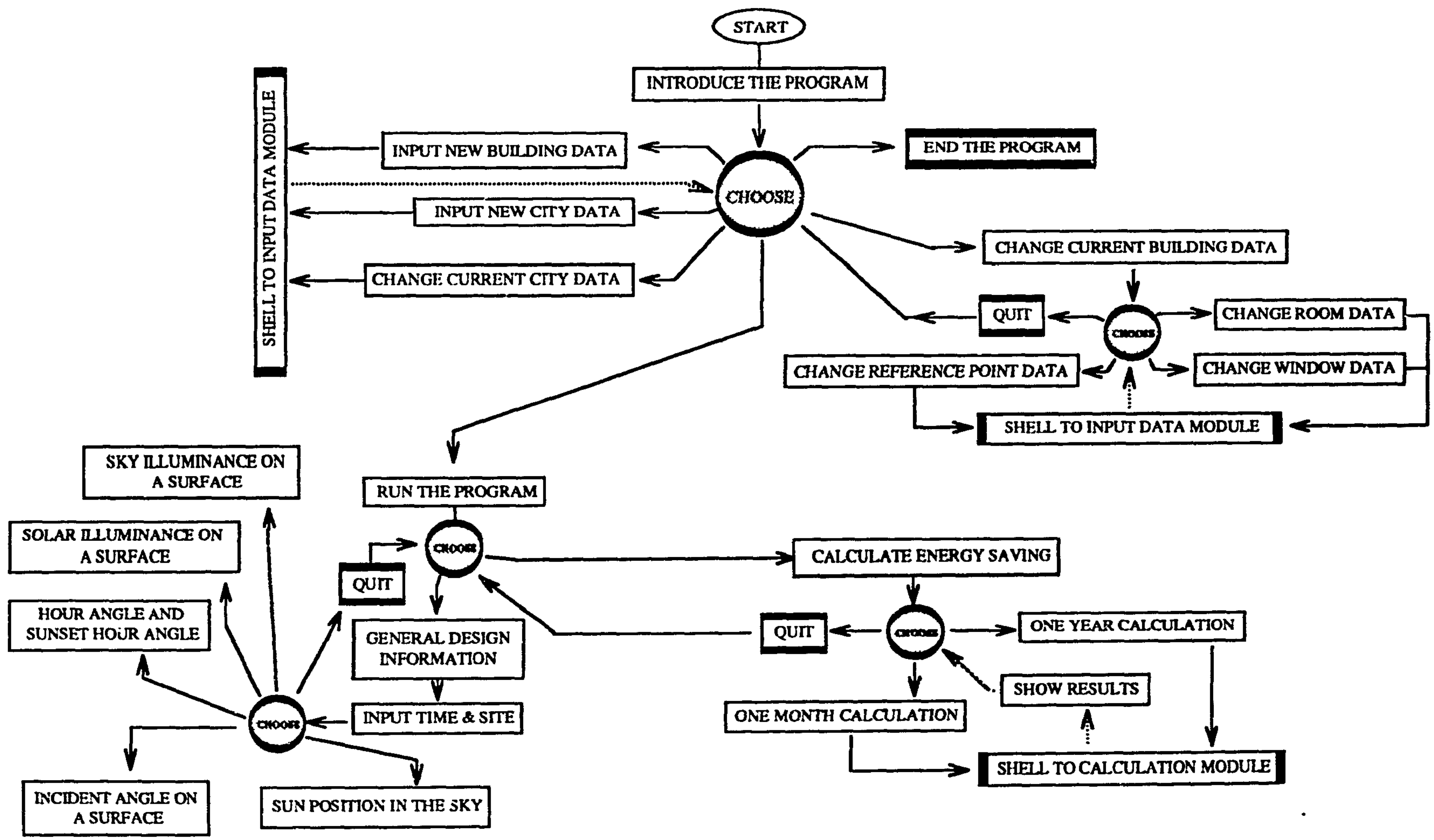


Fig 4.2 Schematic Diagram for Tasks Performed by the Control Module

	1	2	3	4	5	6	7
Month 1	Clear Sky Prob.	P. Cloudy Sky Prob.	Overcast Sky Prob.	Ave. Out Door Temp.	Ave. Horz. Radiation	Clearness Factor	Degree Days
Month 2	Clear Sky Prob.	P. Cloudy Sky Prob.	Overcast Sky Prob.	Ave. Out Door Temp	Ave. Horz. Radiation	Clearness Factor	Degree Days
Month 12	Clear Sky Prob.	P. Cloudy Sky Prob.	Overcast Sky Prob.	Ave. Out Door Temp.	Ave. Horz. Radiation	Clearness Factor	Degree Days
General	Latitude	Longitude	Altitude (Km)	Time zone stand. mer.			

Fig 4.3 Schematic for Weather Data Matrix (WEATH)

	Month 1	Month 2	Month 3							Month 12
General	# of spaces in building	Climate type	Cost of Kw. Hour							
2	# of work. days	# of work. days	# of work. days							# of work. days
3	Cooling or heating month	Cooling or heating month	Cooling or heating month							Cooling or heating month
4	Req. indoor temp.	Req. indoor temp.	Req. indoor temp.							Req. indoor temp.

Fig 4.4 Schematic for Building Data Matrix (BIDA)

RODA (room # , row , column)

row \ column	1	2	3	4	5	6	7
1 Window wall	Length	Inclined Hight	Azimuth angle	Reflectance	# of windows	Inclination angle on horizontal	Area
2 Right wall	Length	Hight	Azimuth angle	Reflectance	# of windows		Area
3 Opposite wall	Length	Hight	Azimuth angle	Reflectance	# of windows		Area
4 Left wall	Length	Hight	Azimuth angle	Reflectance	# of windows		Area
5 Floor	Window wall length	Right / Left wall length		Reflectance			Area
6 Ceiling	Window wall length	Right / Left wall length		Reflectance			Area
7 Room over all	Aver. refl. of the room surfaces	Total walls area	# of windows in the room		# of sensors in the room	Working hours per day	Outside ground reflectance
8 Room over all	Luminairs coeff. of utilization	Lamp light loss factor	Lamp luminous efficacy	Efficiency of light control sys.	Air condit. cooling efficiency	U value for the outside wall	Light Heat Load to Plenum

Fig 4.5 Schematic for Room Data Matrix (RODA)

WIDA (room # , window # , column)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Window # 1	Window's wall # in the room	Value of x	Value of y	Window width (m)	Window hight (f)	Window altitude angle	Window's glass transmittance	Curtain transmittance	Over hang (a) value	Over hang (p) value	Over hang (b) value	Over hang (d) value	Fin (p') value	Fin (b') value	Fin (c') value	Fin (a') value	Glass extinction coefficient	Glass thickness	Window azimuth	Glass conductance value
Window # 2																				

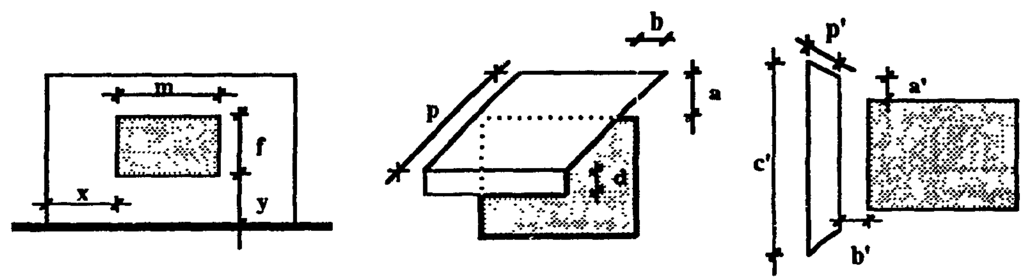
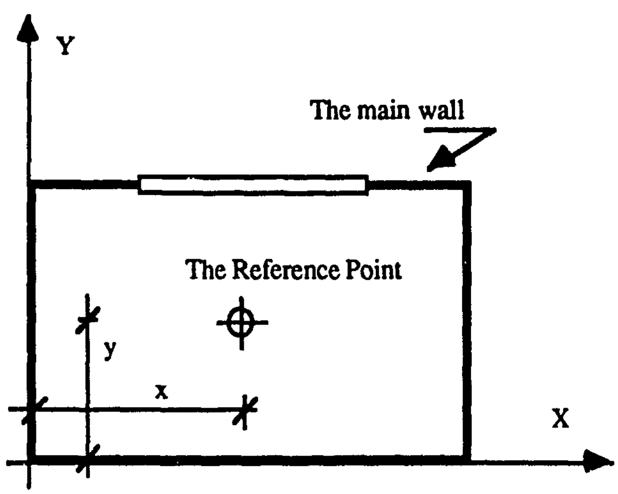


Fig 4.6 Schematic for Window Data Matrix (WIDA)

REFP (room # , sensor # , column)

	1	2	3	4	5
Sensor # 1	x value	y value	z value	Required illumination level	Area covered by the sensor



Plan

Fig 4.7 Schematic for Reference Point Data Matrix (REFP)

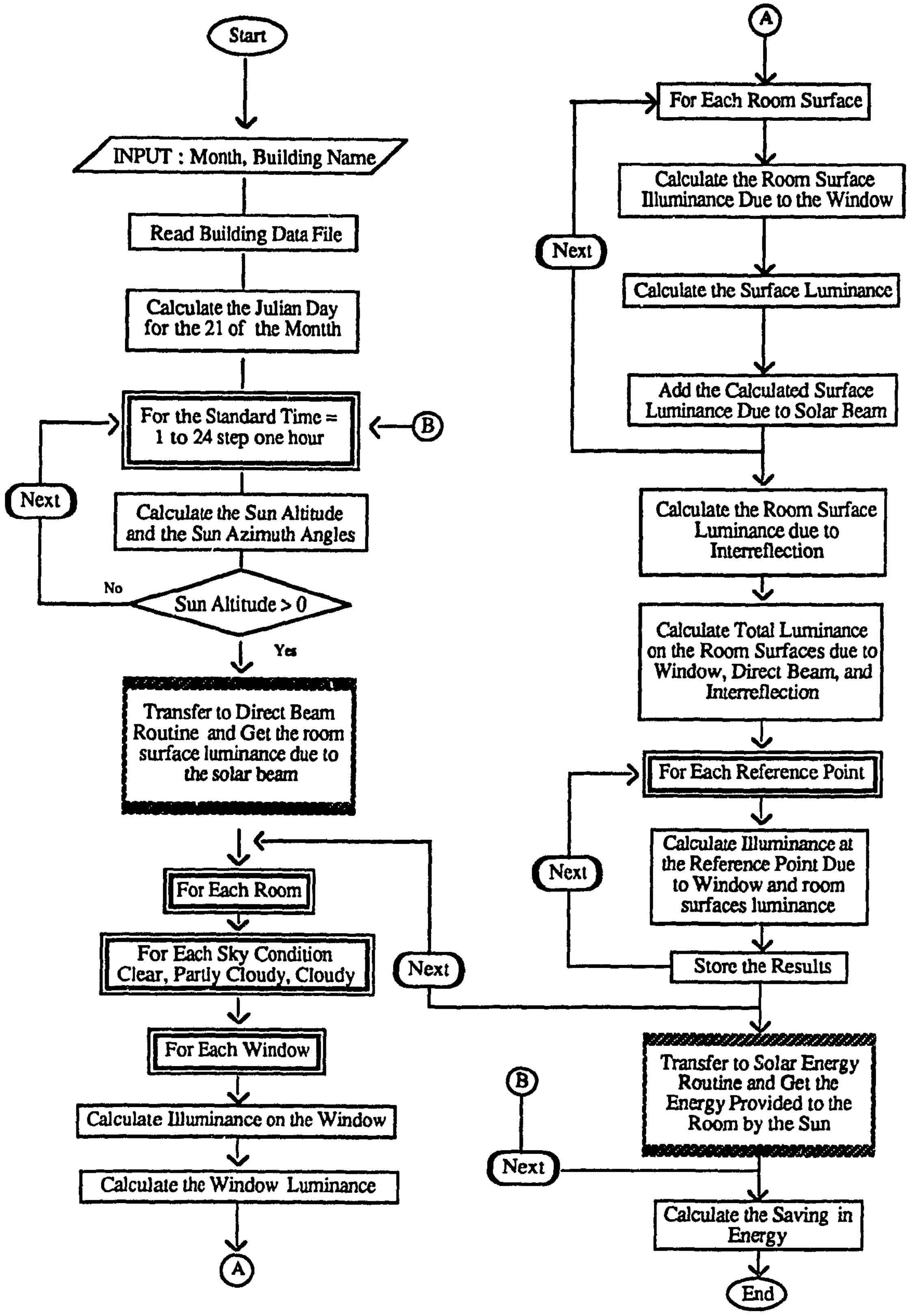


Fig 4.8 Main / Diffuse Sky Illumination Routine

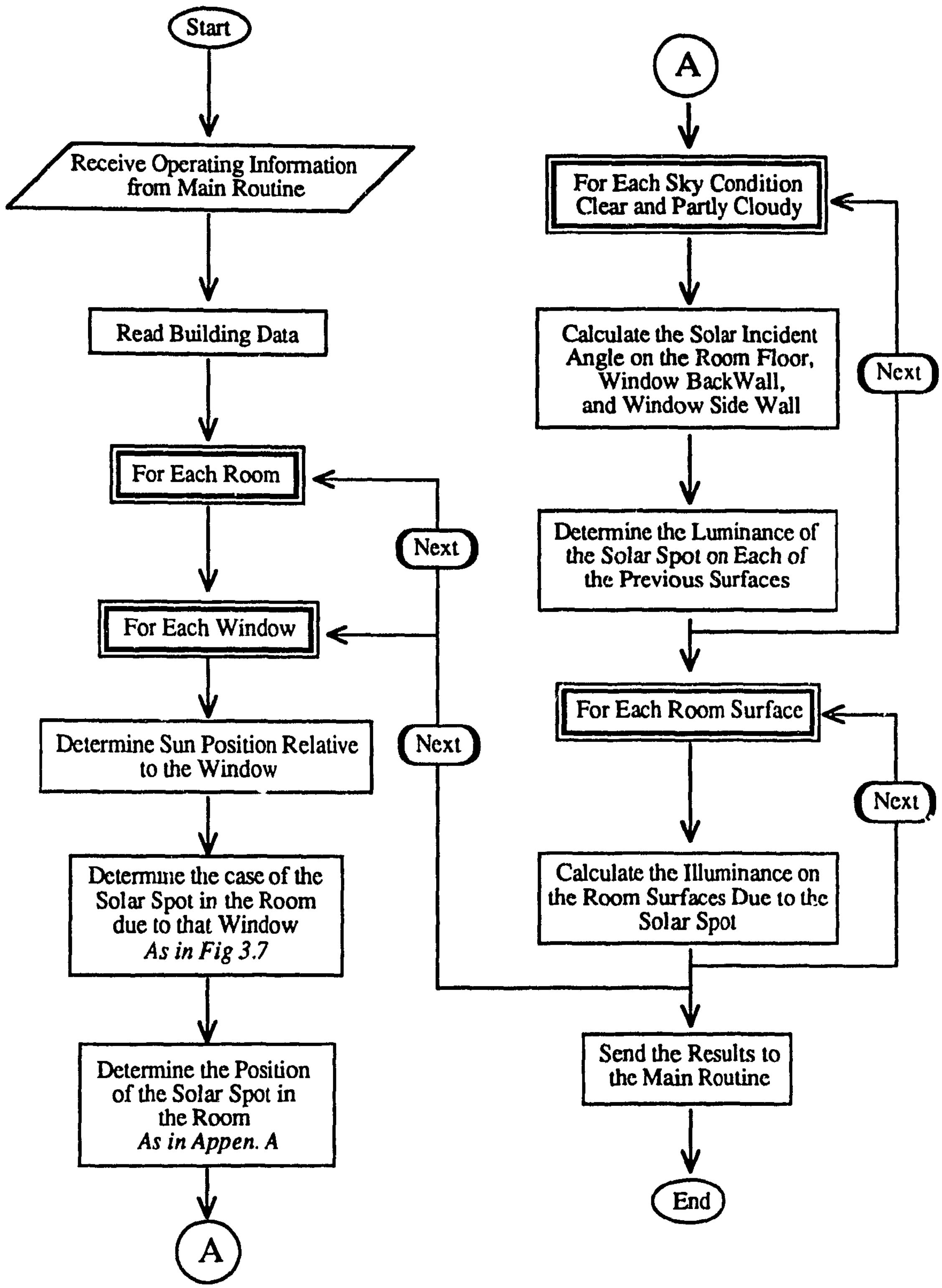


Fig 4.9 Direct Beam Routine

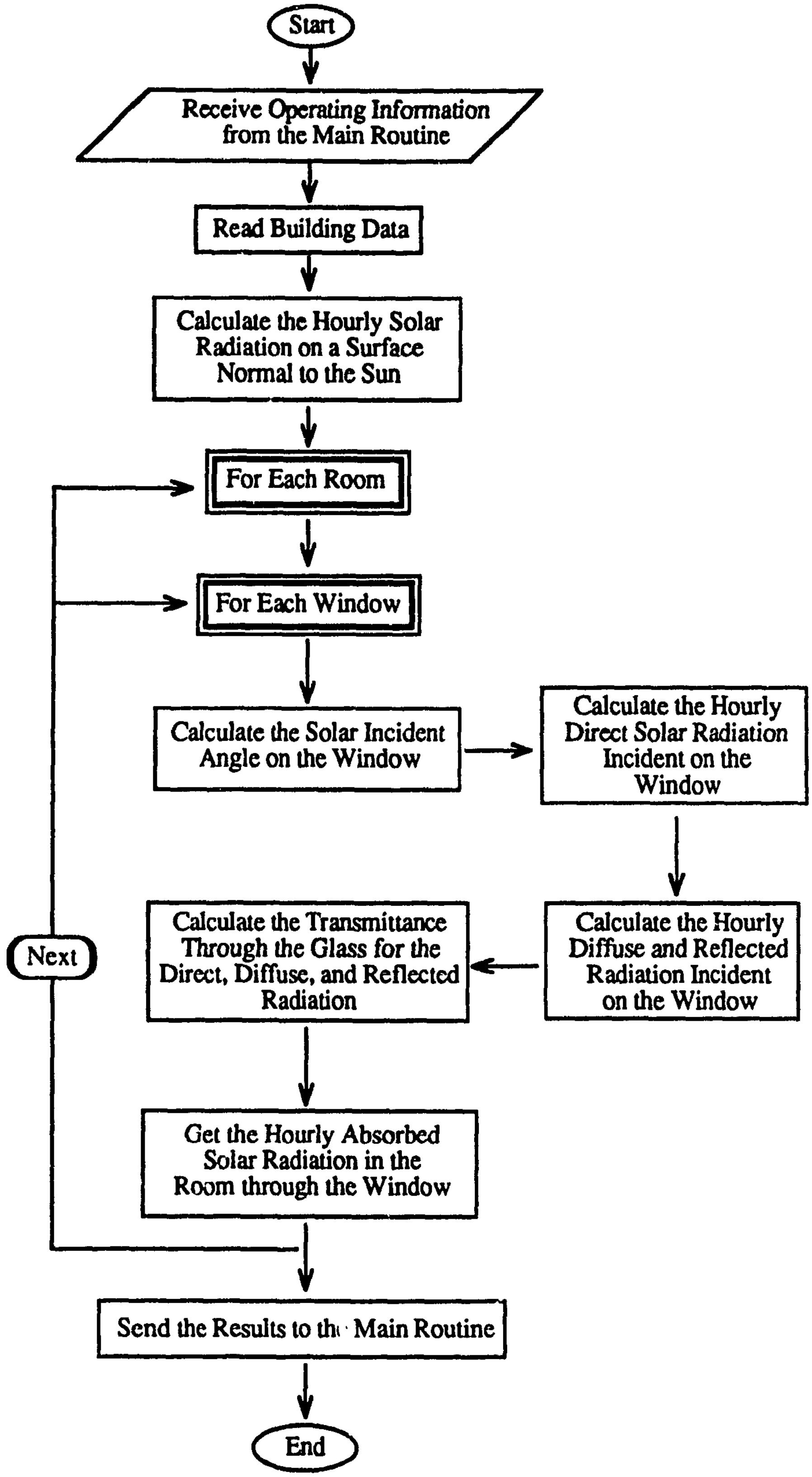
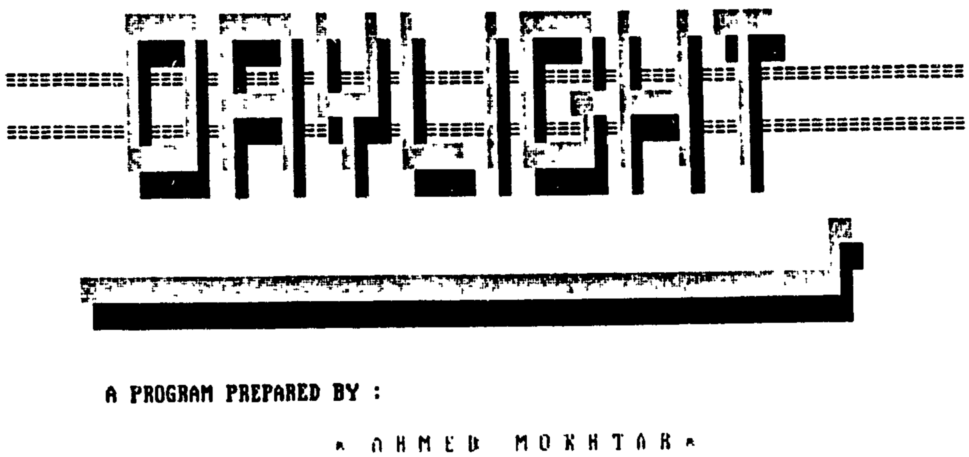


Fig 4.10 Solar Energy Routine



Introduction < *Control Module* >

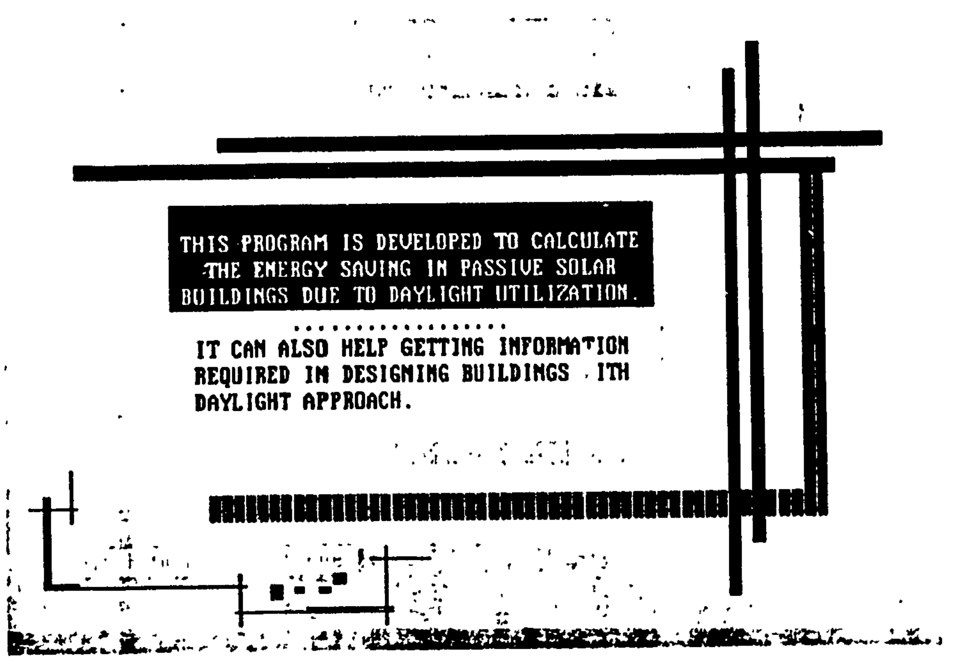
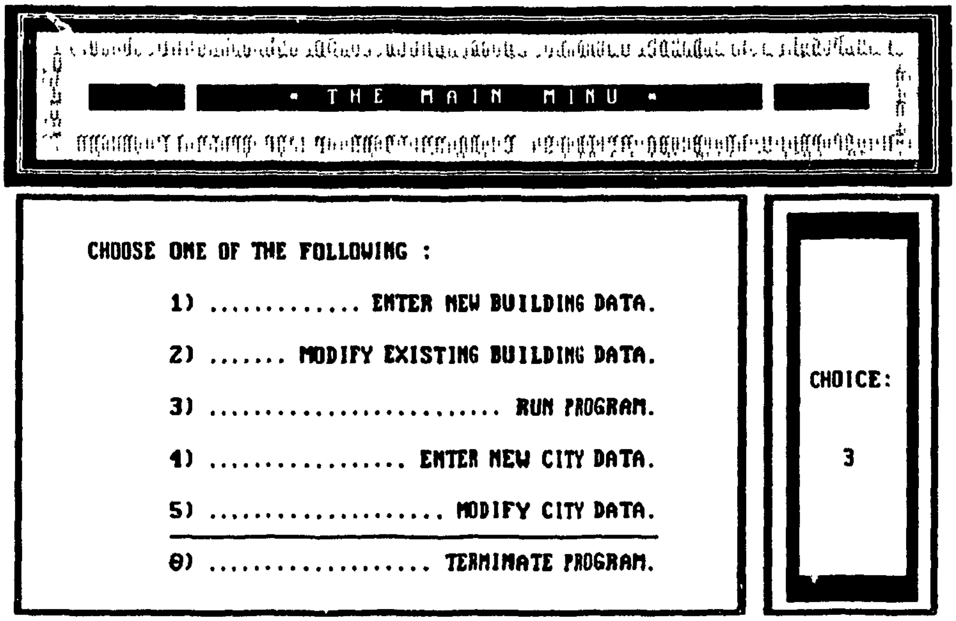
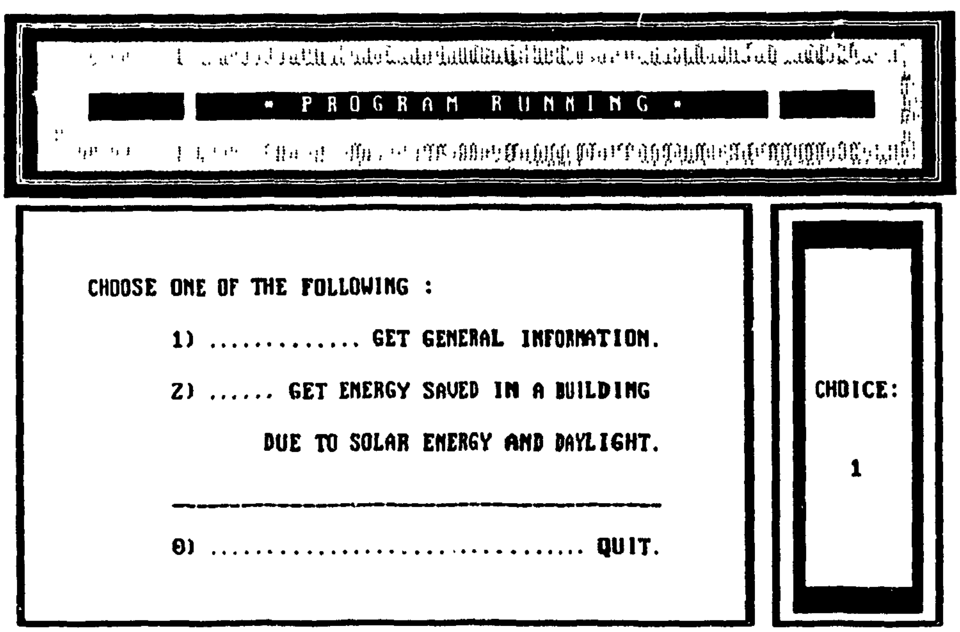
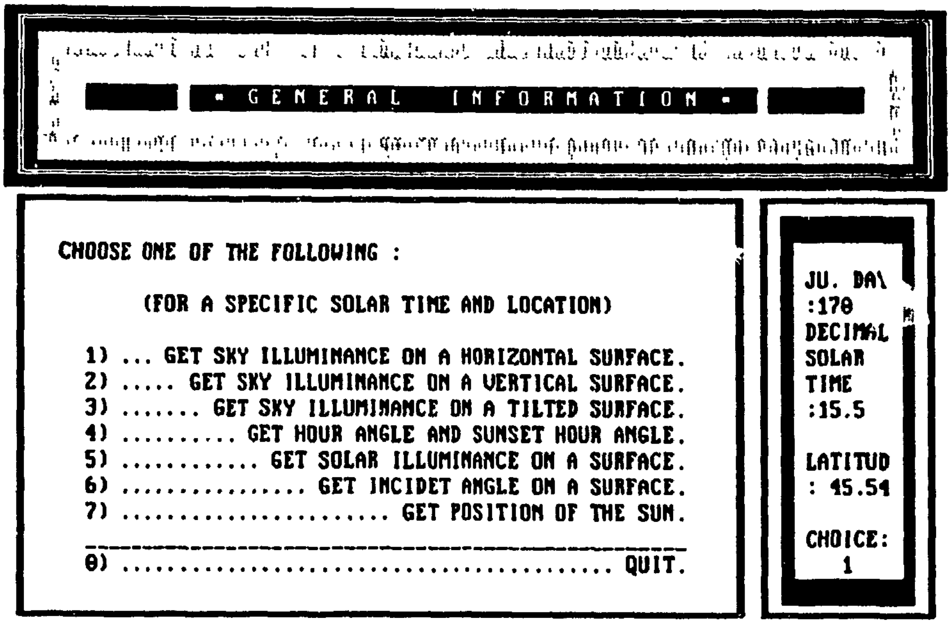


Fig 4.11 A Step by Step Presentation of the Screens Displayed to the User during the Application Example

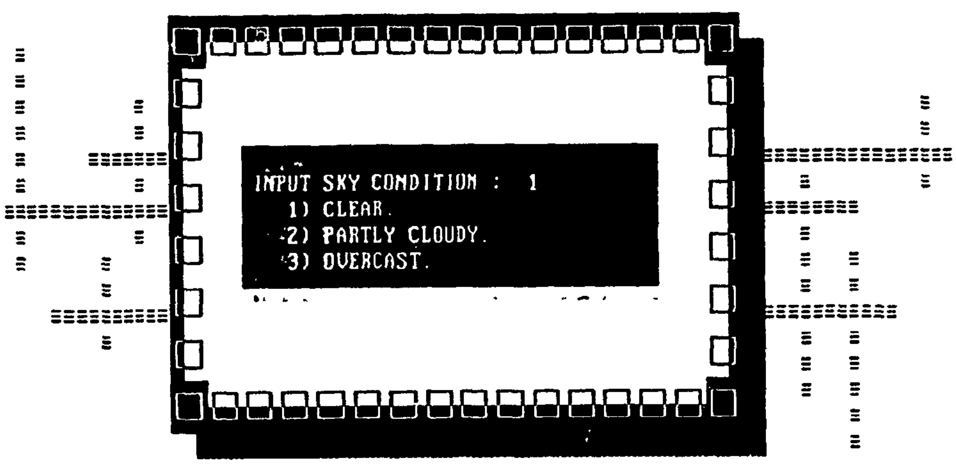


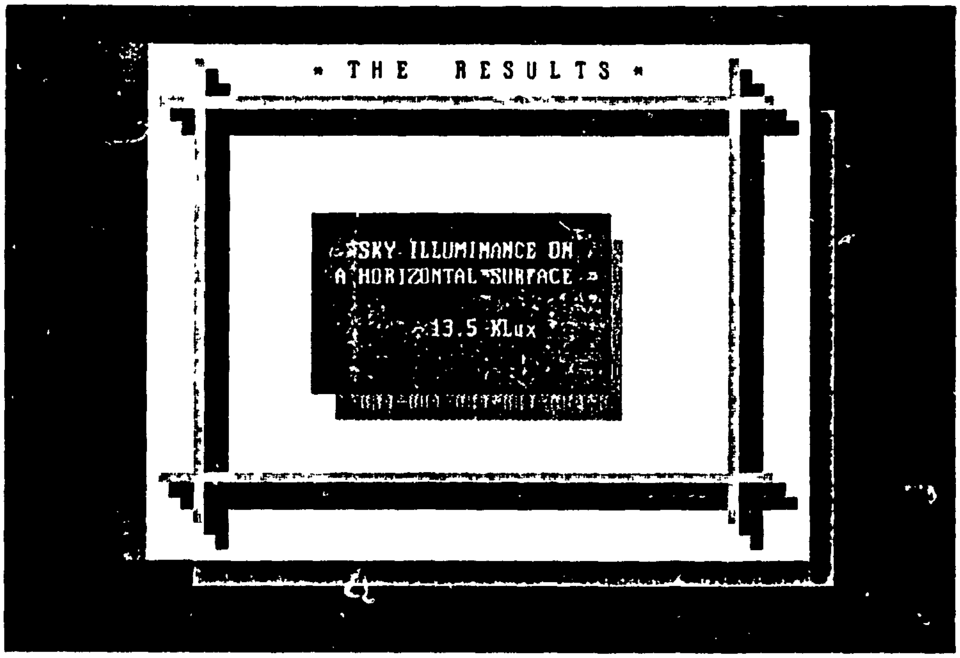
Choosing a Path < *Control Module* >



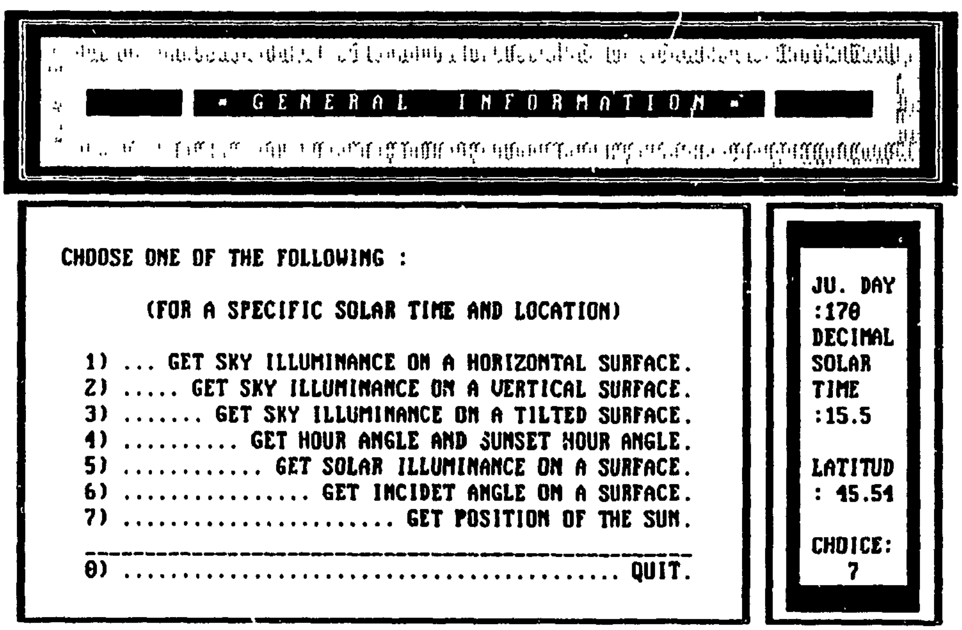


Information Request, Send Operating Orders to the Calculation Module
< *Control Module* >

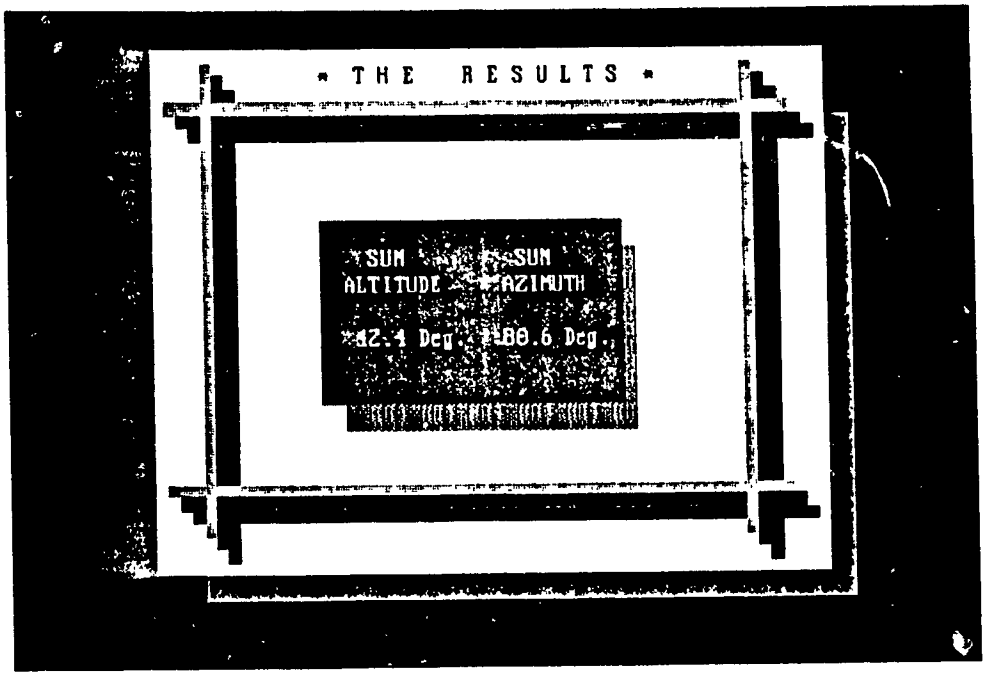




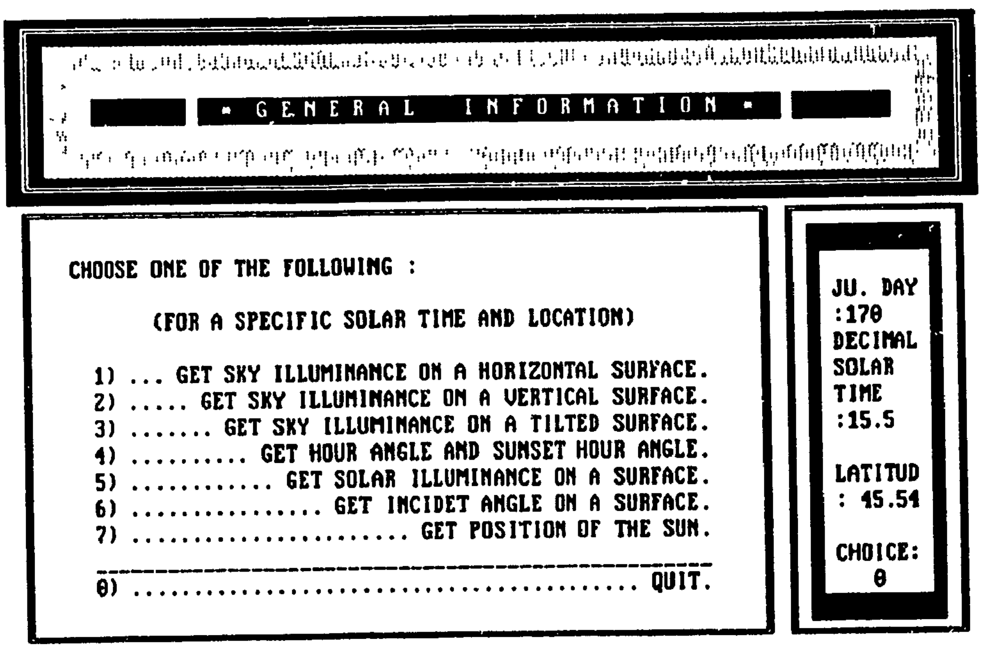
Showing Results Imported from the Calculation Module
< *Control Module* >



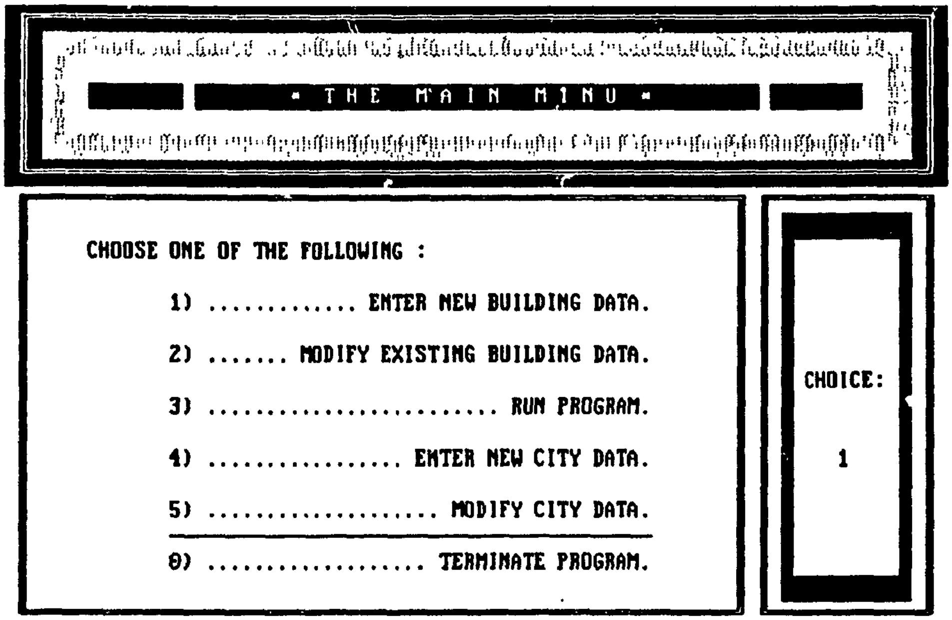
Information Request, Send Operating Orders to
the Calculation Module
< *Control Module* >



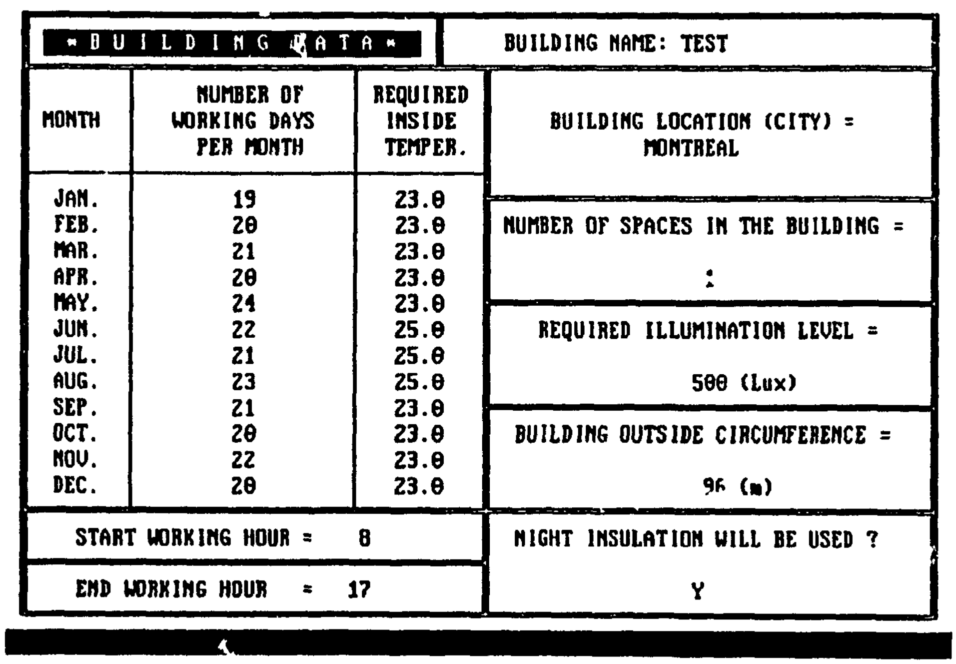
Showing Results Imported from the Calculation Module
< *Control Module* >



Quit This Path < *Control Module* >



Choosing a Path < *Control Module* >



Data Sheet < *Input Data Module* >

*** B U I L D I N G   D A T A ***   BUILDING NAME : TEST

* REFERENCE BUILDING DATA *

REFERENCE BUILDING IS A BUILDING THAT HAS THE SAME DESIGN
OF THE ACTUAL BUILDING EXCEPT THAT ITS SOLAR COLLECTING
SPACES ARE ASSUMED TO HAVE NEUTRAL EXTERNAL WALLS.
(NO FORM of HEAT TRANSFER IS ALLOWED THROUGH THESE WALLS)

ENVELOPE CONDUCTANCE * AREA =	TOTAL EXTERNAL WALLS AREA	WINDOW'S TOTAL CIRCUMFER.	TOTAL ILLUMINATED AREA
974 W/°C	= 532 m²	= 140 m	= 576 m²

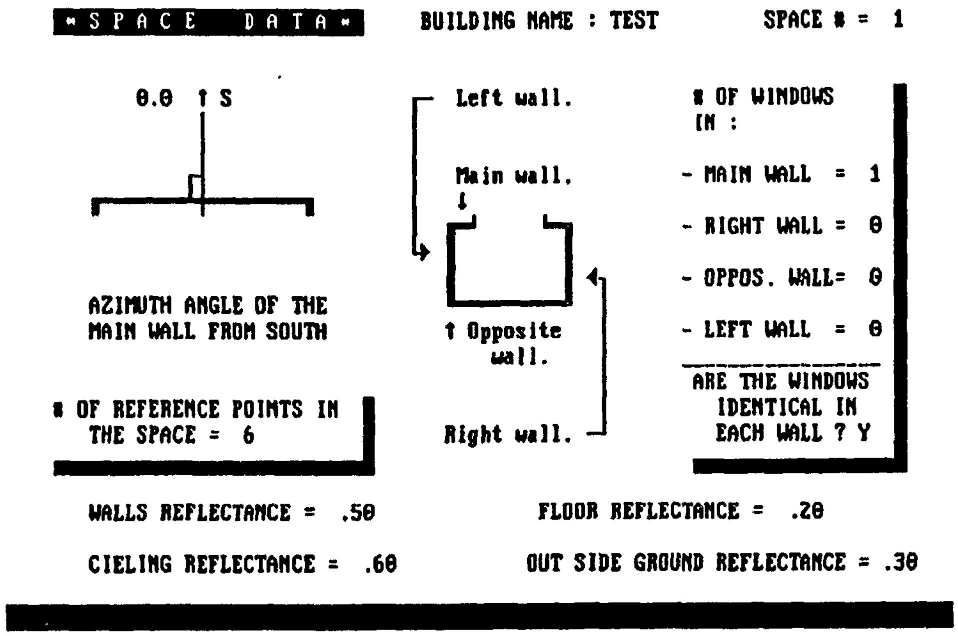
Data Sheet   < ***Input Data Module*** >

*** S P A C E   D A T A ***   BUILDING NAME : TEST   SPACE # = 1

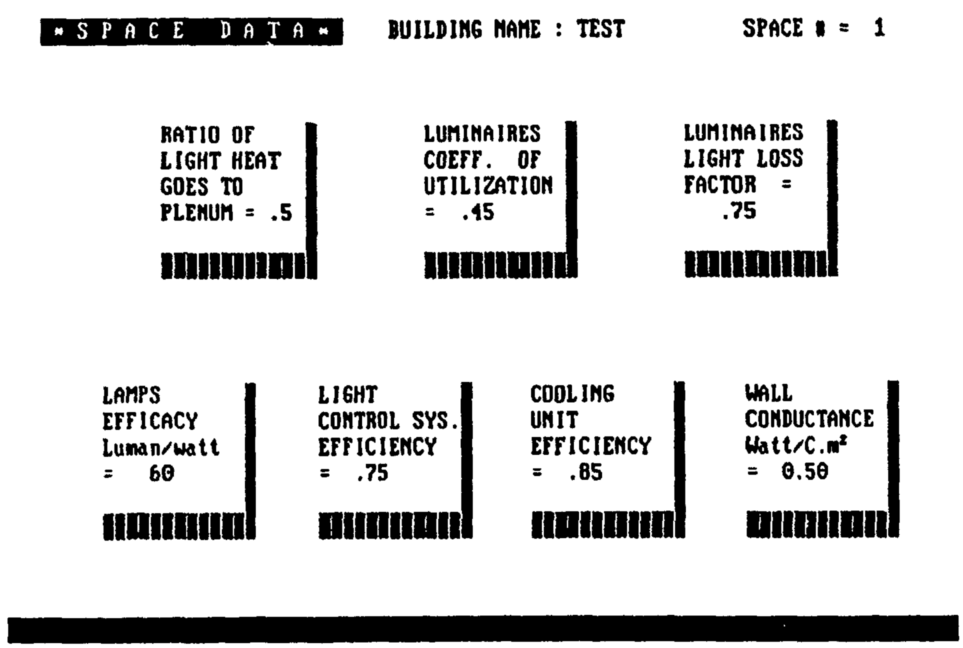
IS THERE AN INCLINED WALL IN
THIS SPACE (Y/N) ? N

" IF Yes THEN YOU CAN
SPECIFY WINDOWS ONLY ON
THIS INCLIND WALL "

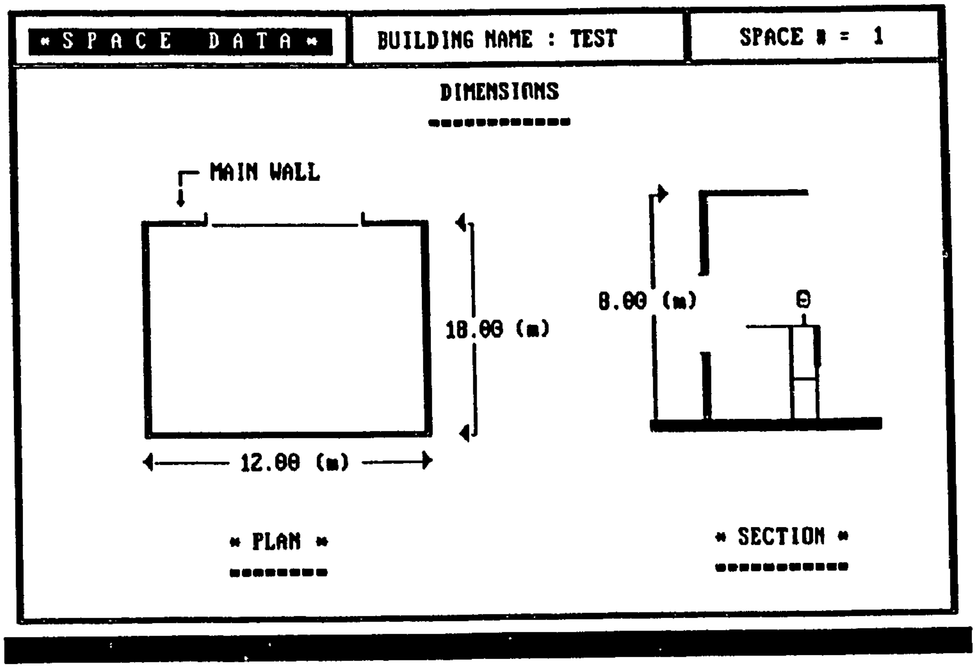
Data Sheet   < ***Input Data Module*** >



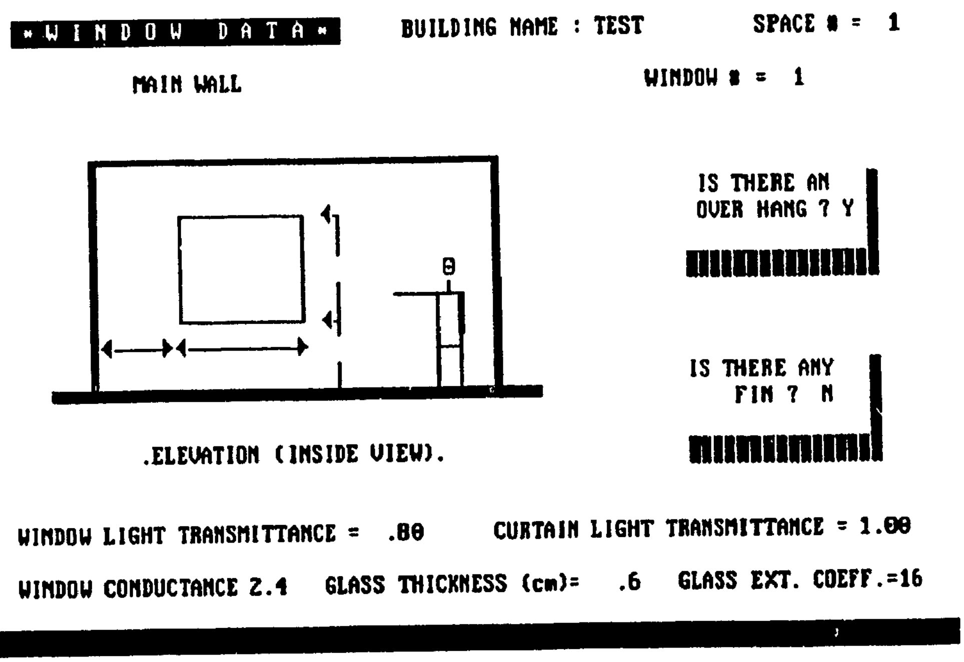
Data Sheet < *Input Data Module* >



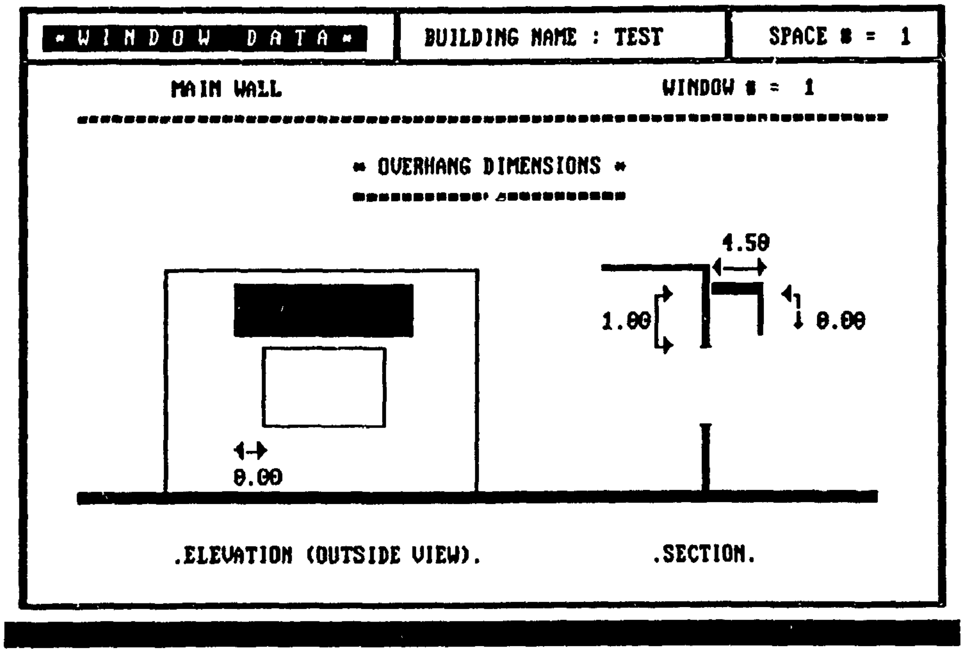
Data Sheet < *Input Data Module* >



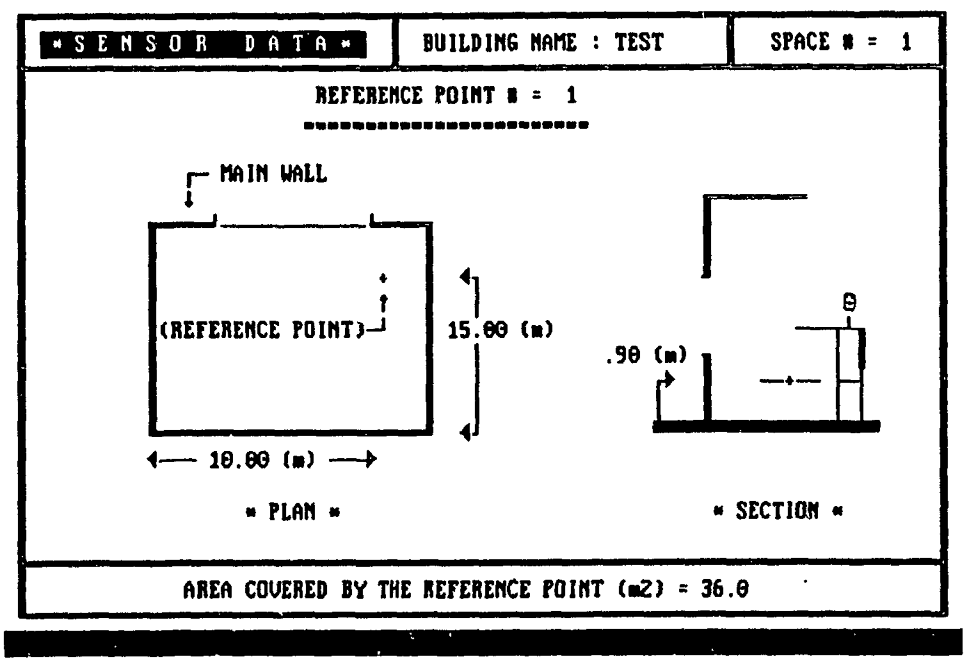
Data Sheet < *Input Data Module* >



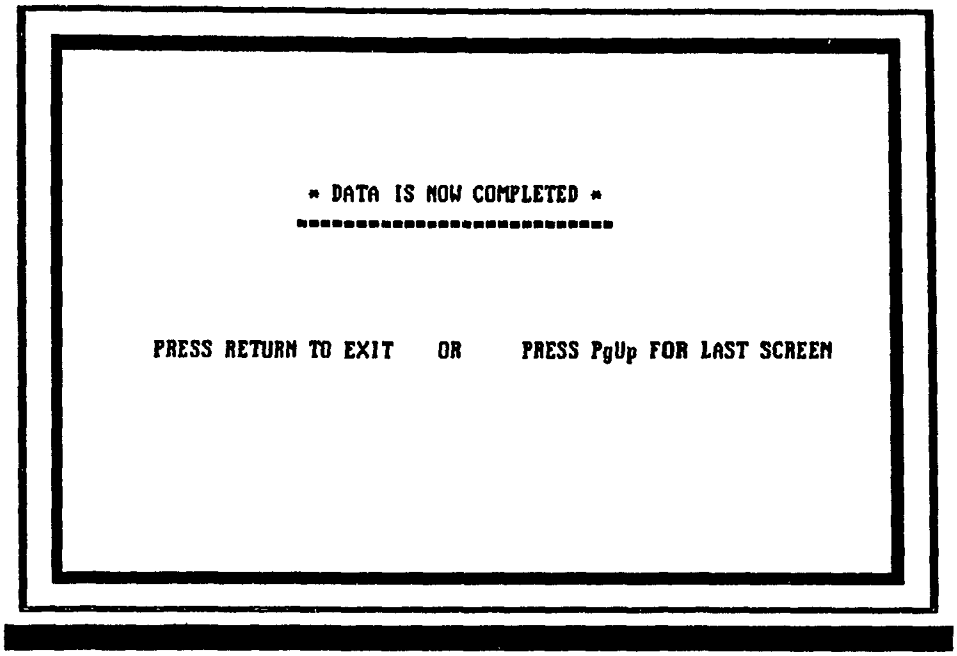
Data Sheet < *Input Data Module* >



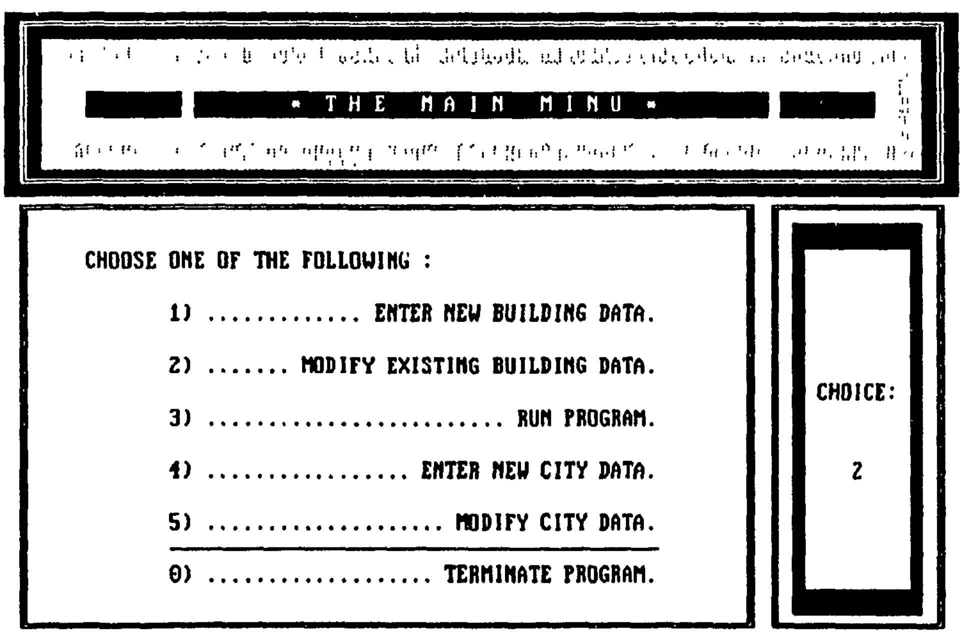
Data Sheet < *Input Data Module* >



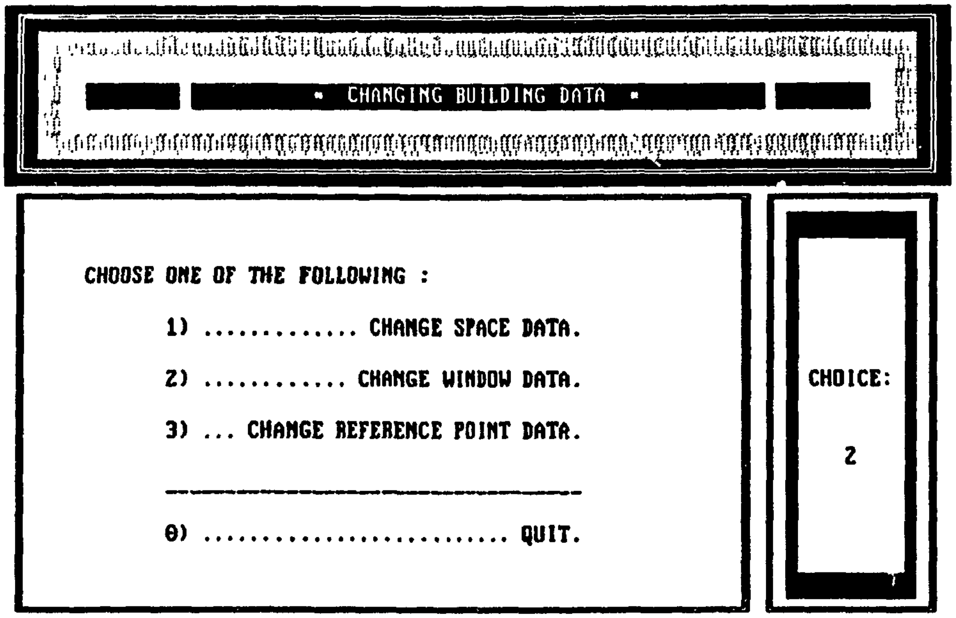
Data Sheet < *Input Data Module* >



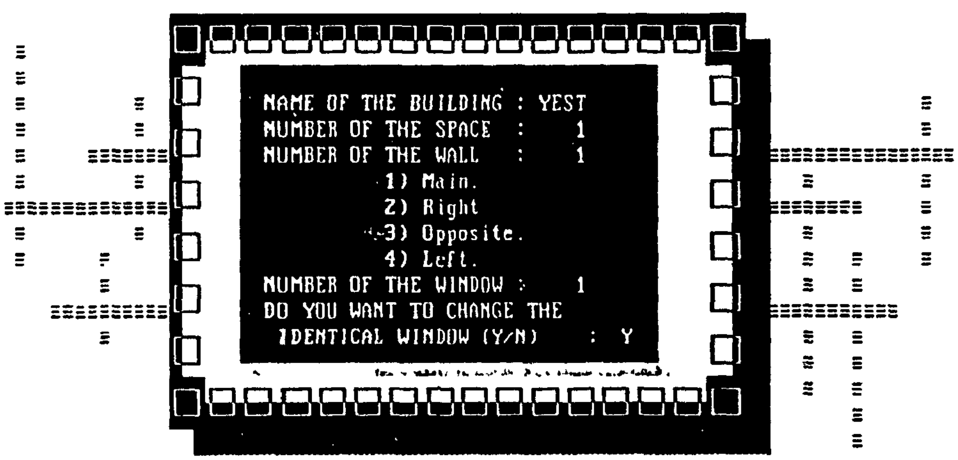
Quit this Module < *Input Data Module* >



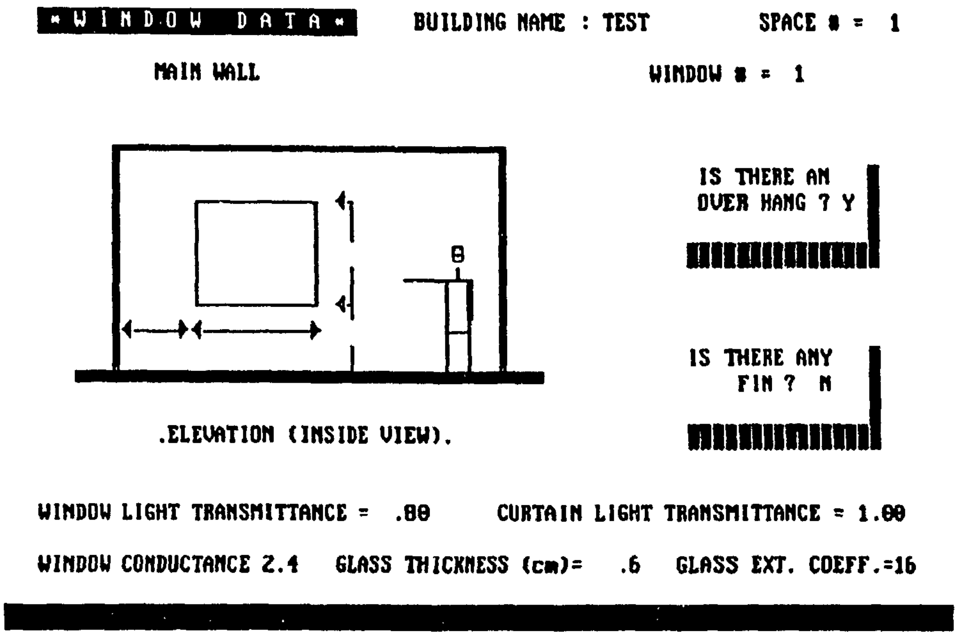
Choosing a Path < *Control Module* >



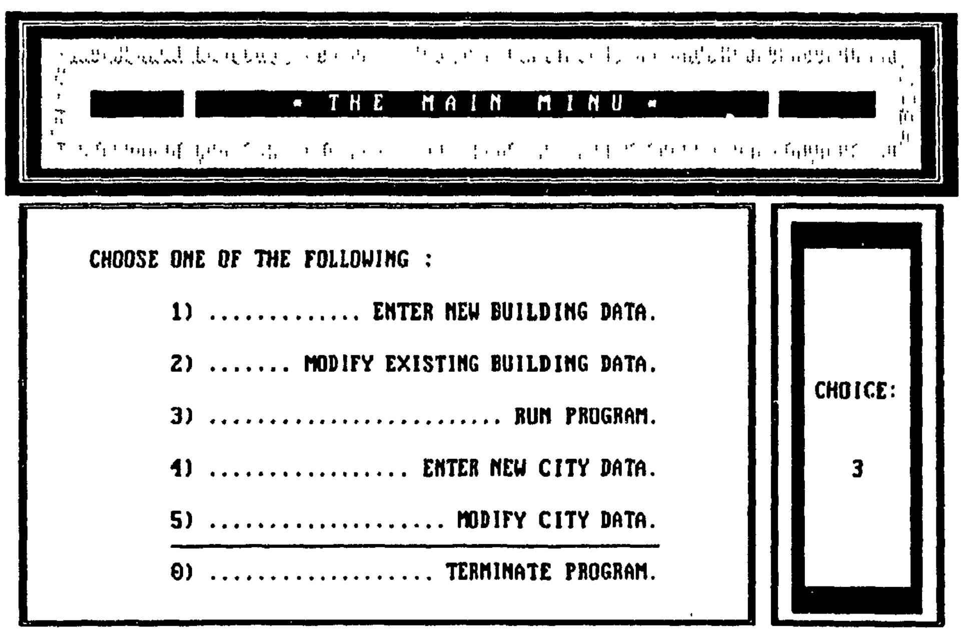
Choosing a Path < *Control Module* >



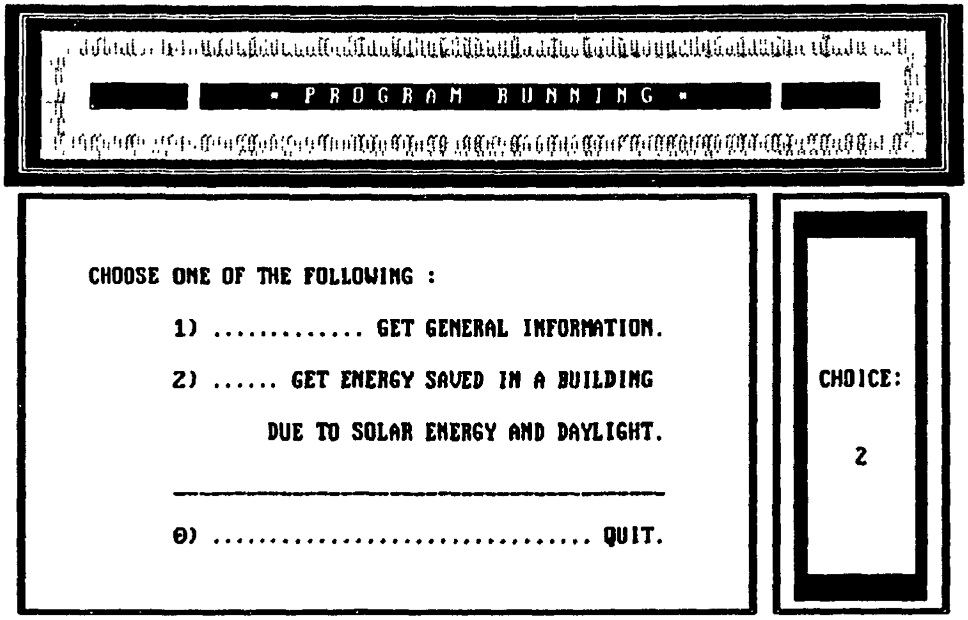
Information Request, Send Operating Orders to
the Input Data Module
< *Control Module* >



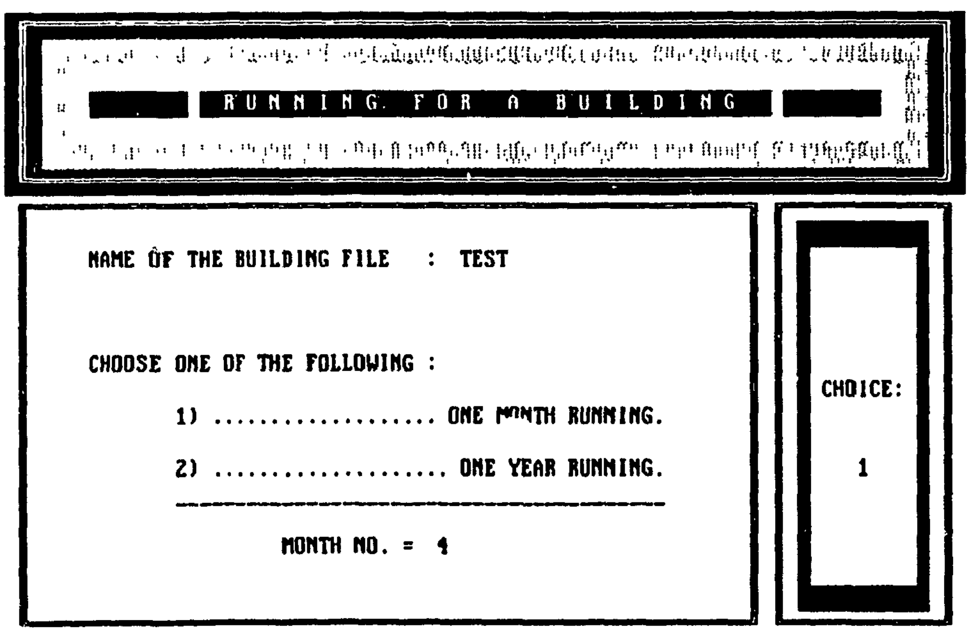
Data Sheet < ***Input Data Module*** >



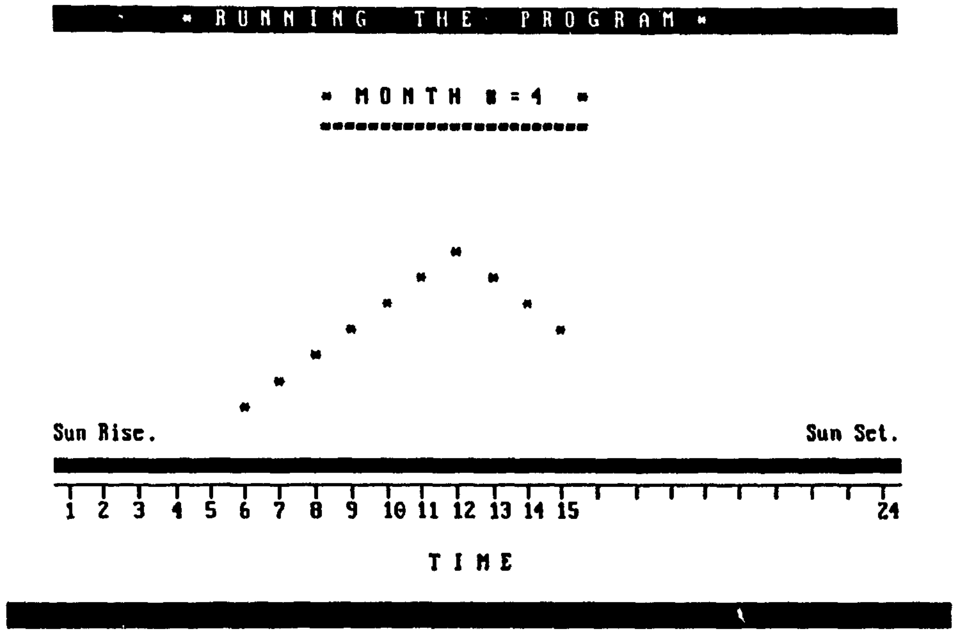
Choosing a Path < ***Control Module*** >



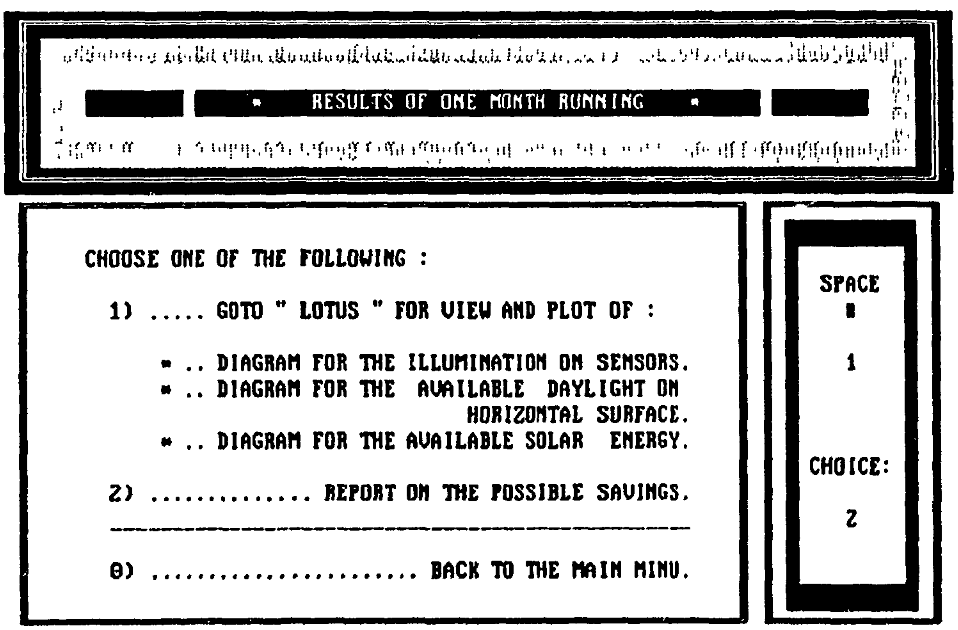
Choosing a Path < *Control Module* >



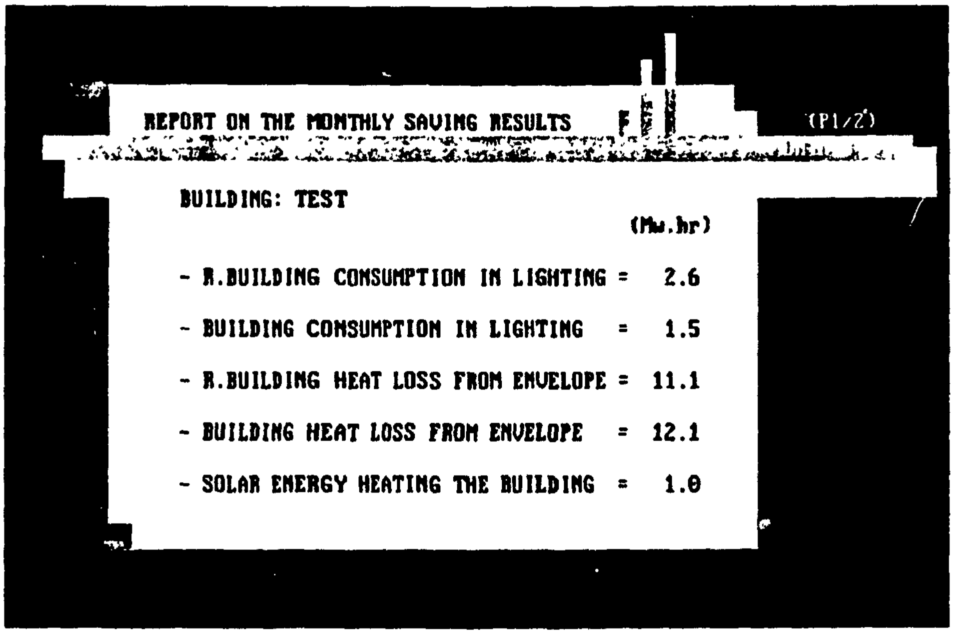
Choosing a Path, Sending Operating Orders to the Calculation Module
< *Control Module* >



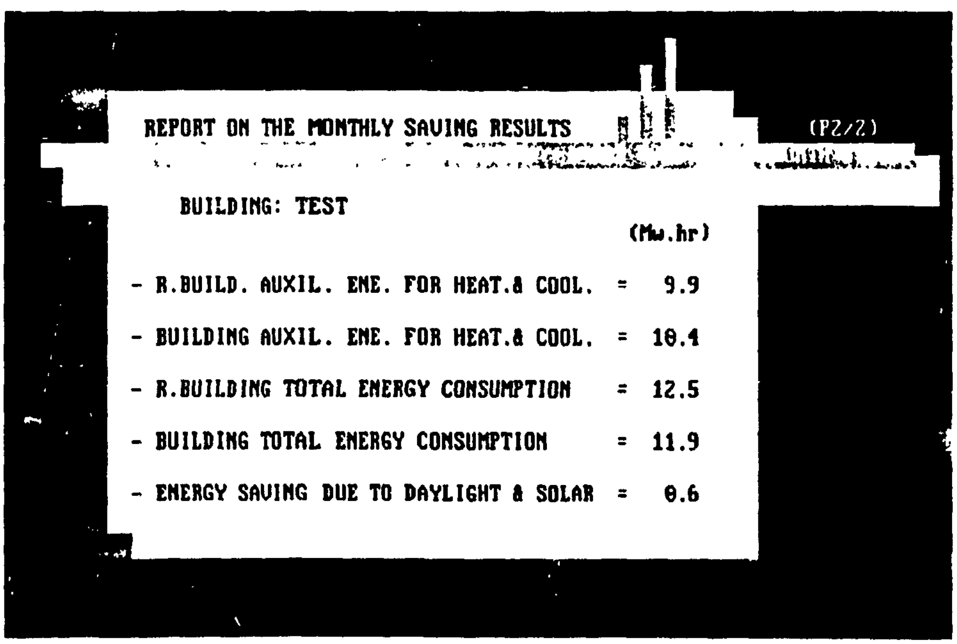
Calculating < ***Calculation Module*** >



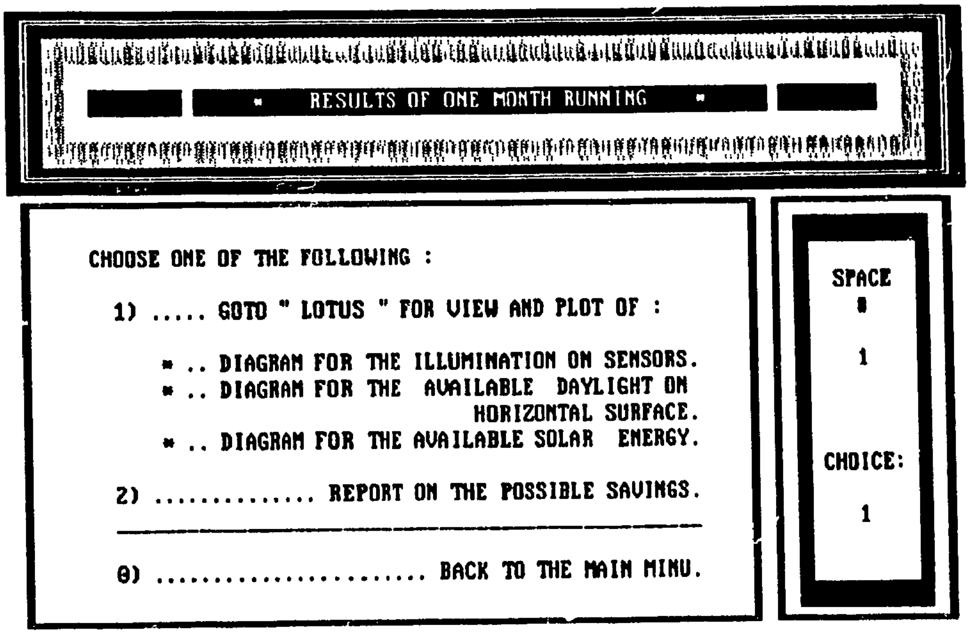
Choosing a Path < ***Control Module*** >



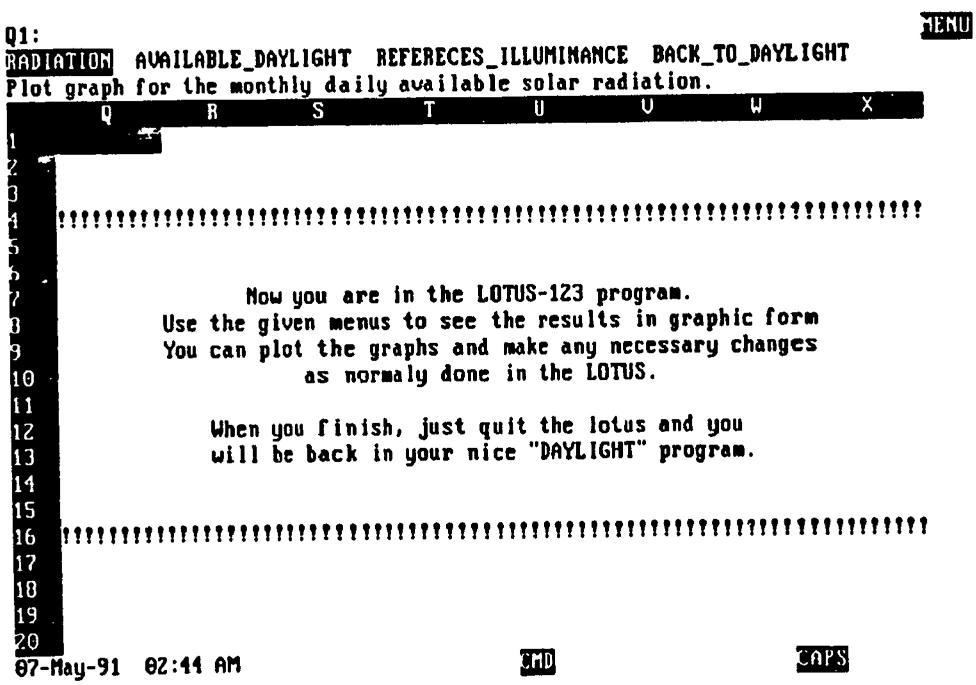
Showing Results Imported from
the Calculation Module
< Control Module >



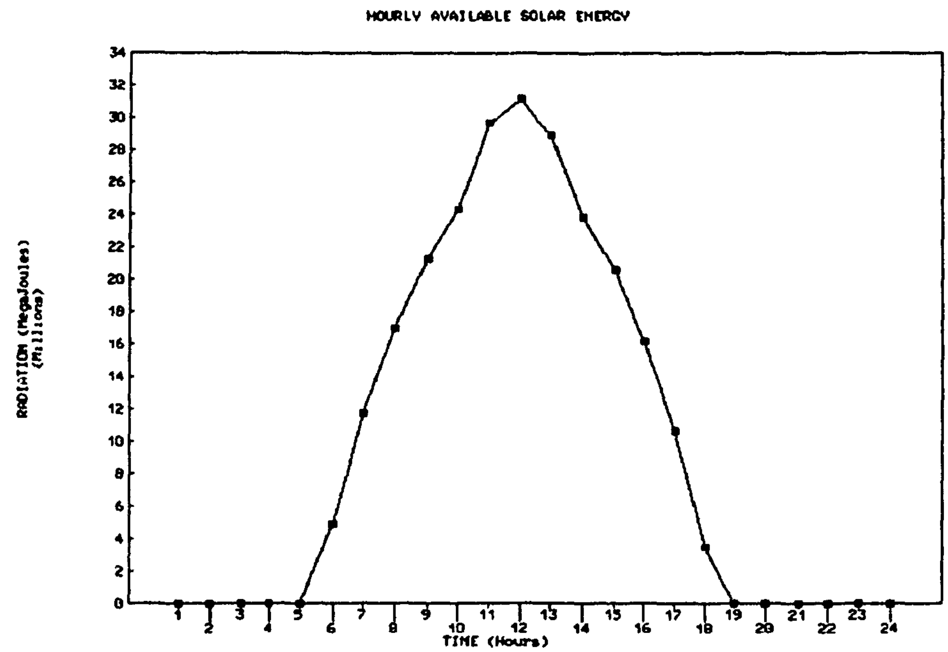
Showing Results Imported from
the Calculation Module
< Control Module >



Choosing a Path *< Control Module >*



Starting Costumized Lotus-123,
Choosing a Path
< Lotus Program >



Showing Results < *Lotus Program* >

MENU

RADIATION AVAILABLE_DAYLIGHT REFERECES_ILLUMINANCE BACK_TO_DAYLIGHT
Plot graphs for the available daylight in different sky conditions.

09-May-91 02:30 AM CMD

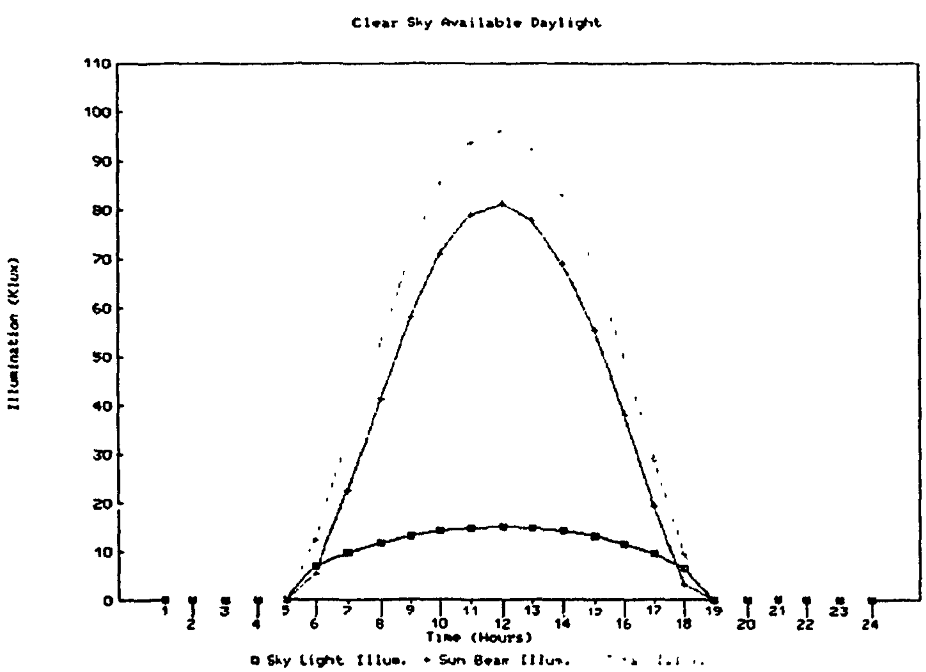
Choosing a Path < *Lotus Program* >

MENU

CLEAR_CHART PARTLY_CHART OVERCAST_CHART MAIN_MENU
Show graph for the available daylight in clear sky condition

09-May-91 02:32 AM CMD

Choosing a Path < ***Lotus Program*** >



Show Results < ***Lotus Program*** >

MENU

RADIATION AVAILABLE_DAYLIGHT REFERECES_ILLUMINANCE BACK_TO_DAYLIGHT
Plot graphs for the illuminance on the room sensors.

09-May-91 02:34 AM CMD

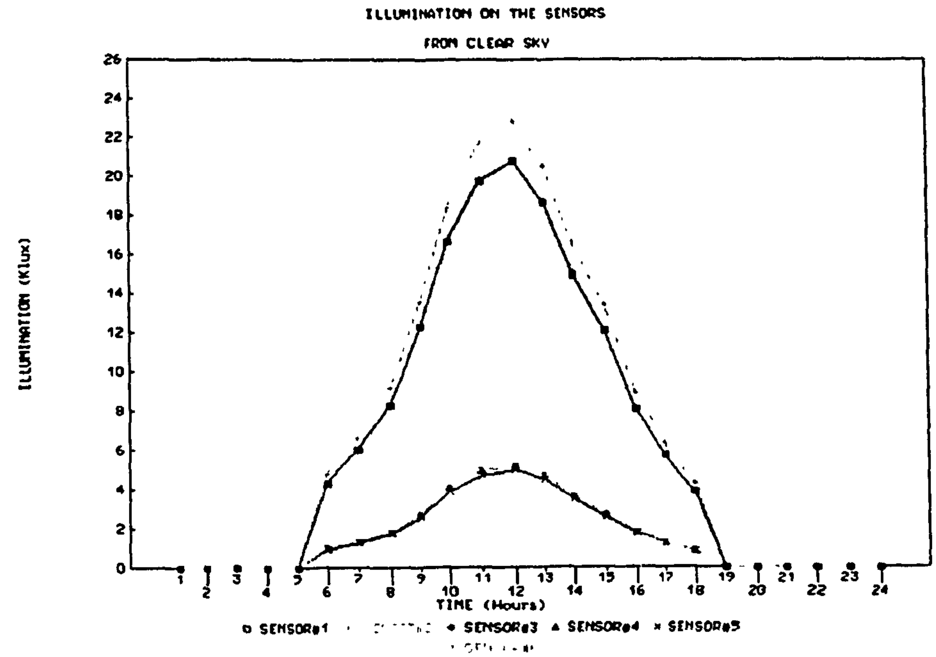
Choosing a Path < *Lotus Program* >

MENU

CLEAR_CHART PARTLY_CHART OVERCAST_CHART MAIN_MENU
Show graph for the sensors illuminance in clear sky condition

09-May-91 02:36 AM CMD

Choosing a Path < *Lotus Program* >

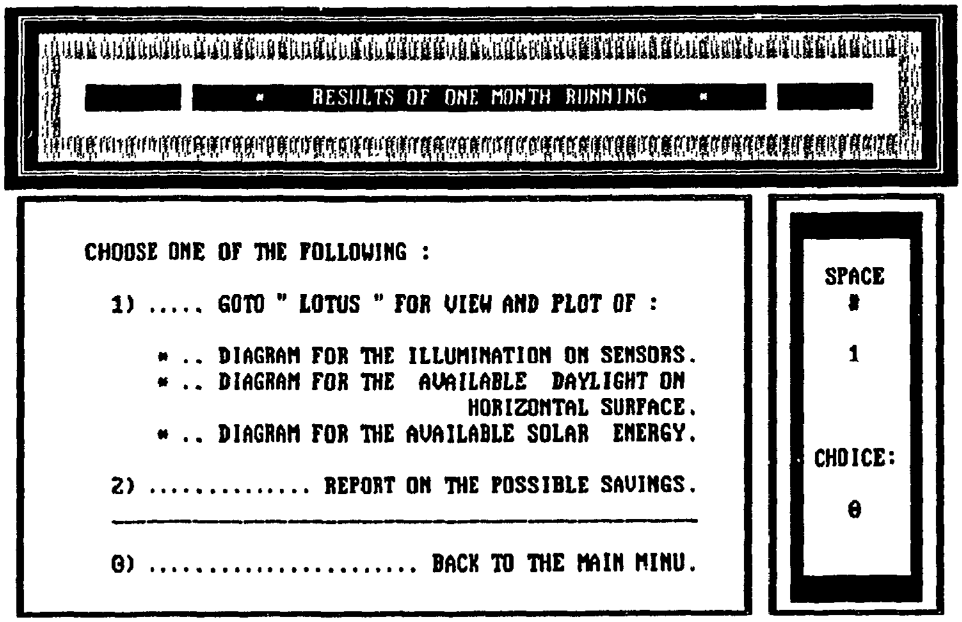


Show Results *< Lotus Program >*

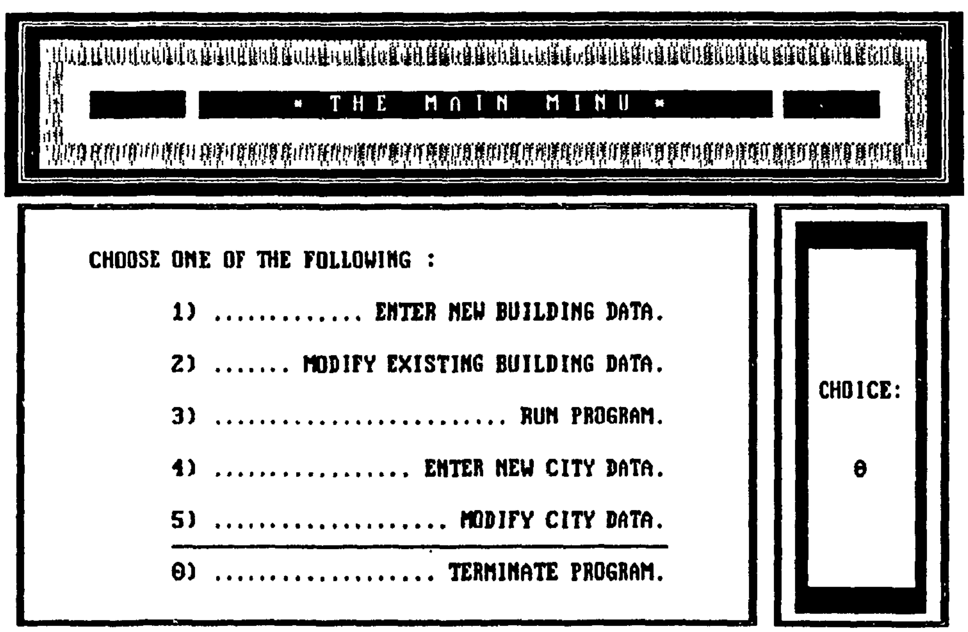
```
                                                                       MENU
RADIATION  AVAILABLE_DAYLIGHT  REFERECES_ILLUMINANCE  BACK_TO_DAYLIGHT
End LOTUS-123
```

```
09-May-91  02:37 AM                      CMD
```

Quit the Lotus-123 *< Lotus Program >*



Choosing a Path < *Control Module* >



Terminate Program < *Control Module* >

CHAPTER V

MODEL VALIDITY AND EVALUATION

5.1. APPROACH

The present model is based on previously validated components found in literature. In addition to the development made herein, the selected components found in the literature were modified to suit the purpose of the present model. The quantification of the available daylight on vertical and horizontal surfaces, as well as, the determination of the solar heat gain through a window are calculated using recommended equations by the IES and ASHRAE successively. These equations are previously validated and consequently it will not be evaluated. On the other hand, the equations for quantifying daylight on tilted surfaces are available from invalidated sources. Locating the sunspot in the room and determining the total illuminance on the reference points are dependent on the calculations developed in this thesis. These parts will be the subject of this evaluation.

5.2. DIFFUSE SKY ILLUMINANCE ON TILTED SURFACES

This model has utilized the equations given by Nettleton and Murdoch [47] to determine the available daylight on a tilted surface. No measurements have been done to validate these equations in Montreal. The instrument shown in Fig. (5.1) was designed to conduct these measurements. It consists of two plates (a base and a reference plane) which are connected together by four adjustable screws to ensure the horizontality of the reference plane. A circular rotating plate is connected to the reference plane and it is free to rotate horizontally to change the azimuth angle of the measuring surface. Two vertical rods are fixed to the rotating plate and connected together with an axis. The axis is fixed to the measuring plate and allow it to rotate vertically to change its tilt angle. The axis is also connected to a guiding protractor that allows it to rotate in an increment of 15°. A small shadow band is connected to an adjustable carrier so as to protect the light sensor on the measuring surface from direct sun light. Daylight was measured using this instrument on the shore of St. Lawrence river at Lasalle district in Montreal. That place allowed measuring the sky illuminance without having any significant obstructions. During the months of January and February 1991, two overcast days, two partly cloudy days, and two clear days were measured. The measurements were done in different hours throughout these days. The azimuth angle of the measuring surface was changing between the main eight angles (S, SE, E, NE, N, NW, W, SW) while the surface tilt angle was changing from horizontal to vertical every 15°. The data was collected in sheets as the one shown in Fig. (5.2). Same corresponding days and sky conditions were input into the model and the results were generated and compared with those measured. Fig. (5.3) to Fig. (5.5) shows the comparison between the calculated and the measured results. The figures show that the measured values are higher than those calculated in both the clear and overcast sky conditions. That can be related to the snow covering the ground around the measuring instrument. For the partly cloudy sky, the results are better distributed around the perfect correlation line. That is mainly due to the variability of the cloud ratio and the cloud distribution through the measuring period.

5.3. SUN SPOT LOCATION ALGORITHM

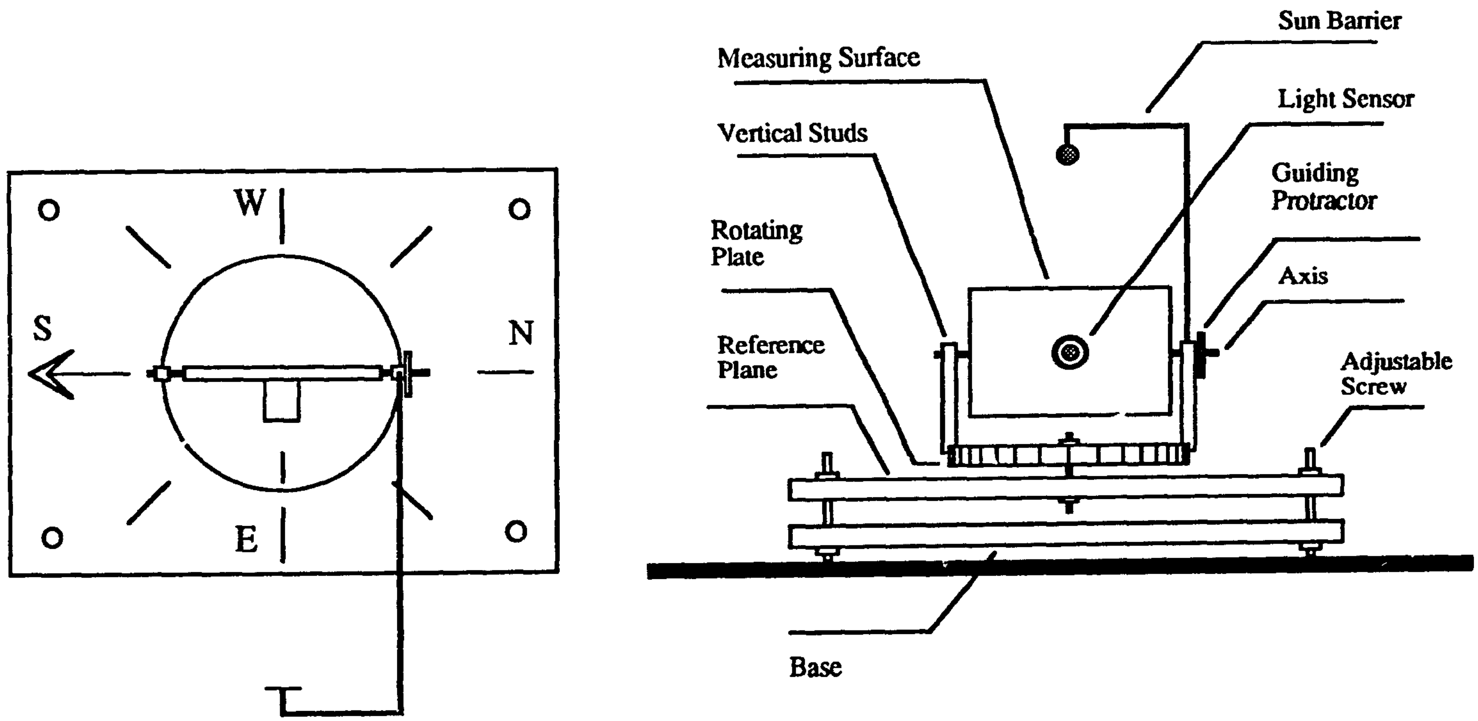
This algorithm will be also examined by comparison with experimental measurements. A room with the dimensions given in Fig. (5.6) is located on the roof of the Center for Building Studies. The room has one relatively large window that is having an azimuth angle of 14° from the south to the east. The window has a small overhang with 12 cm in depth. During clear days in January and March, the location of the solar spot in the room has been observed and classified into the cases shown in Table 3.2. The room data was also fed into the model and the solar spot position cases were determined in the same corresponding times of observation. The comparison between the calculated and the measured results showed total agreement which confirms the creditability of the sun spot location algorithm.

5.4. CALCULATING THE ILLUMINANCE ON THE REFFRENCE POINTS

The reference points receive illuminance flux from the window and each point of the room walls. It is assumed in the model that the window has a uniform luminance which is not the real condition. The model also divides each room surface into three strips and assumes that each strip has a homogeneous luminance. That is different from the actual situation where each point in the room surface has its own luminance. These assumptions will lead to some error in the calculation process. This part measures the significance of that error. Using the same room shown on Fig. (5.6), three light sensors have been installed as shown in Fig. (5.7). The first one was put vertically on the window to measure the daylight received by the window. The second and the third light sensors were put horizontally inside the room. The three sensors were connected to a data acquisition instrument. The window was covered from outside with a translucent white paper to reduce the effect of the surrounding buildings on the measurements and to help distributing as much as possible the available daylight uniformly on the room window. The reflectance of the room internal surfaces were assessed and were found to be 0.5 for the walls, 0.6 for the ceiling, and 0.15 for the floor. Measurements were taken for two separated days during the month of December 1990 while the sky condition was totally overcast. The measured illuminance values on the two horizontal light sensors were divided over those measured on the vertical light sensor on the window to get the reference point illuminance factor. The resulting numbers will reflect the effect of the room surfaces on the light sensors final illuminance. The room data and the light sensors positions were fed to the model and the ratios between the calculated window luminance and the calculated illuminance on the corresponding reference points were determined. The results of both the experiment and the model were compared as shown in Fig. (5.8). The figure shows higher calculated values than the measured ones. That can be partly due to the existence of some furniture in the room during the measuring procedures. It is also due to the error in the calculating procedures.

5.5. CONCLUSION

The experimental work has indicated that the assumptions and the equations used in the model give fairly close results to the real conditions. These results give confident in the suggested model and the developed computer software.



Plan

Section

Fig 5.1 The Design of the Instrument Used to Measure Daylight on Tilted Surfaces

SKY CONDITION CLEAR (WITHOUT SUN), PARTLY CLOUDY, OVER CAST ________ DAY ________

HOUR ___

HORIZONTAL

	NORTH	NORTH EAST	EAST	SOUTH EAST	SOUTH	SOUTH WEST	WEST	NORTH WEST
TILT 30								
TILT 45								
TILT 60								
TILT 75								
VERTICAL								

HOUR ___

HORIZONTAL

	NORTH	NORTH EAST	EAST	SOUTH EAST	SOUTH	SOUTH WEST	WEST	NORTH WEST
TILT 30								
TILT 45								
TILT 60								
TILT 75								
VERTICAL								

Fig 5.2 Sky Illumination Data Sheet.

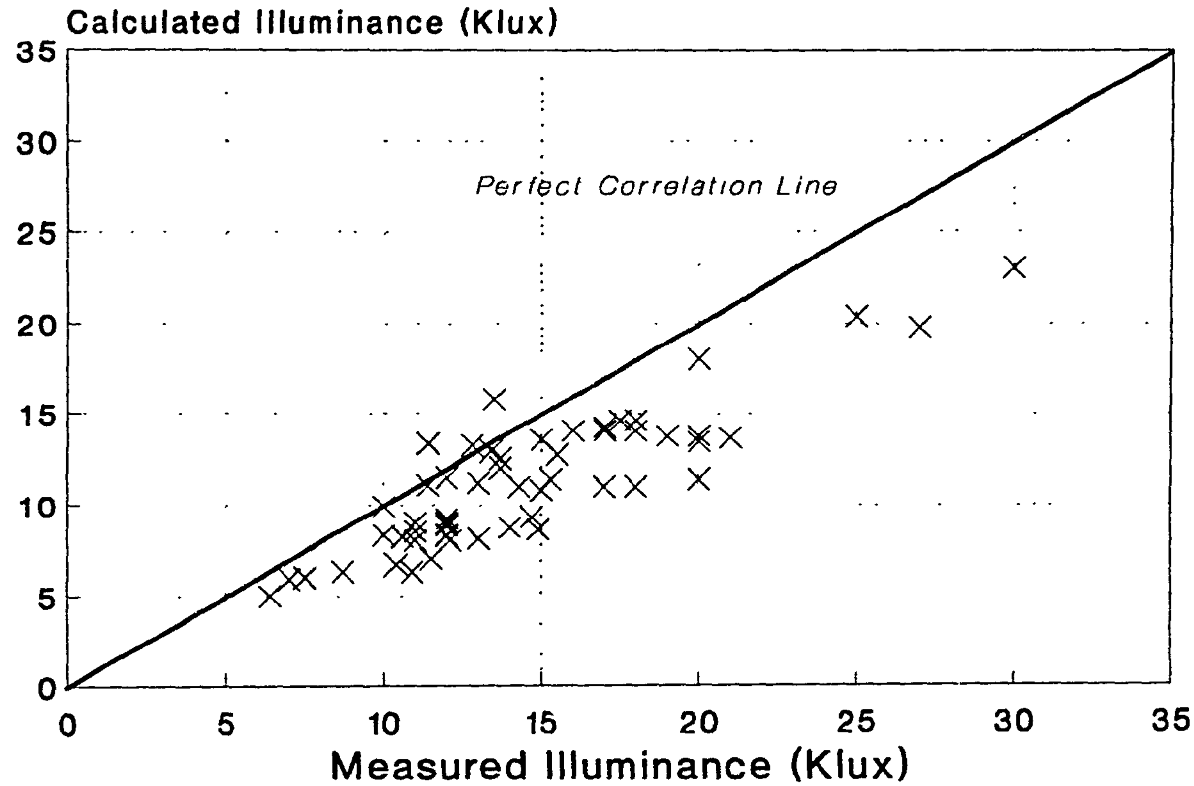


Fig 5.3 Measured Versus Calculated Illuminance on Tilted Surfaces for Clear Sky.

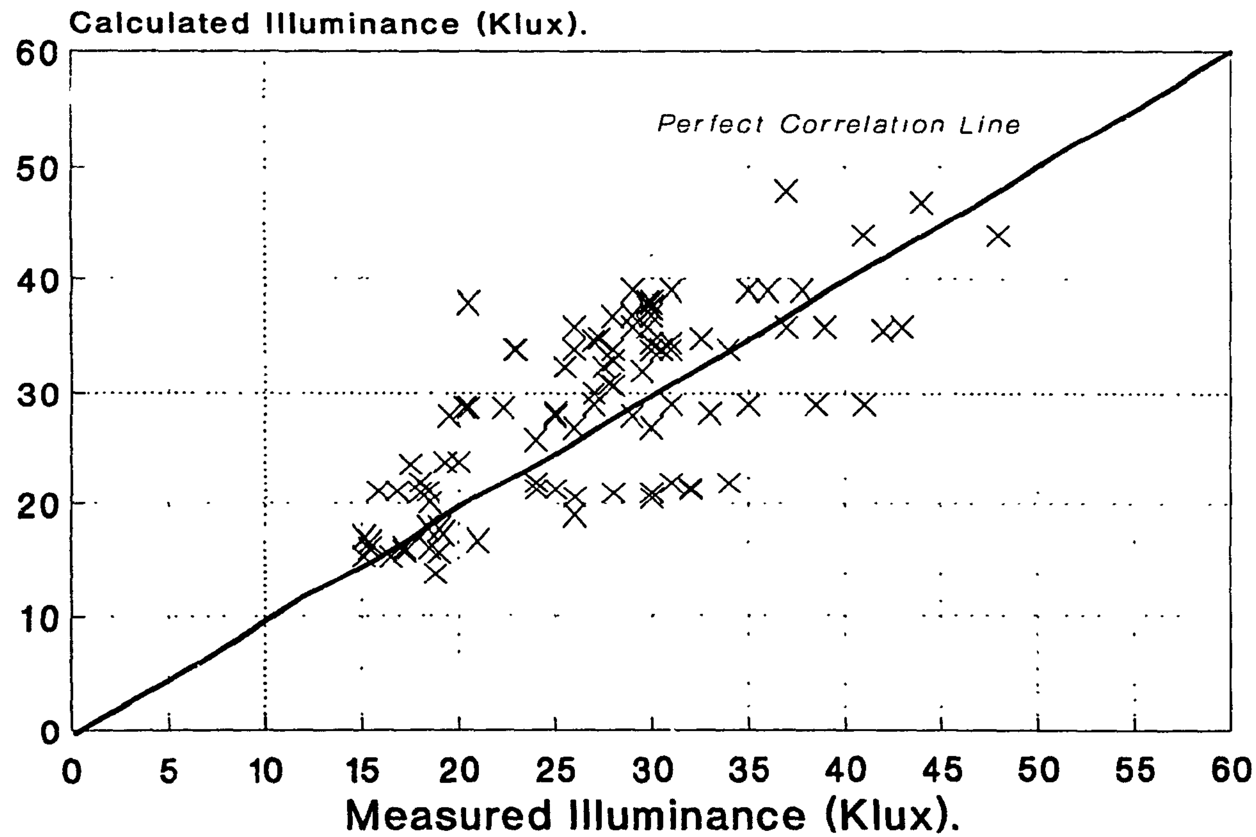


Fig 5.4 Measured Versus Calculated Illuminance on Tilted Surfaces for Partly Cloudy Sky.

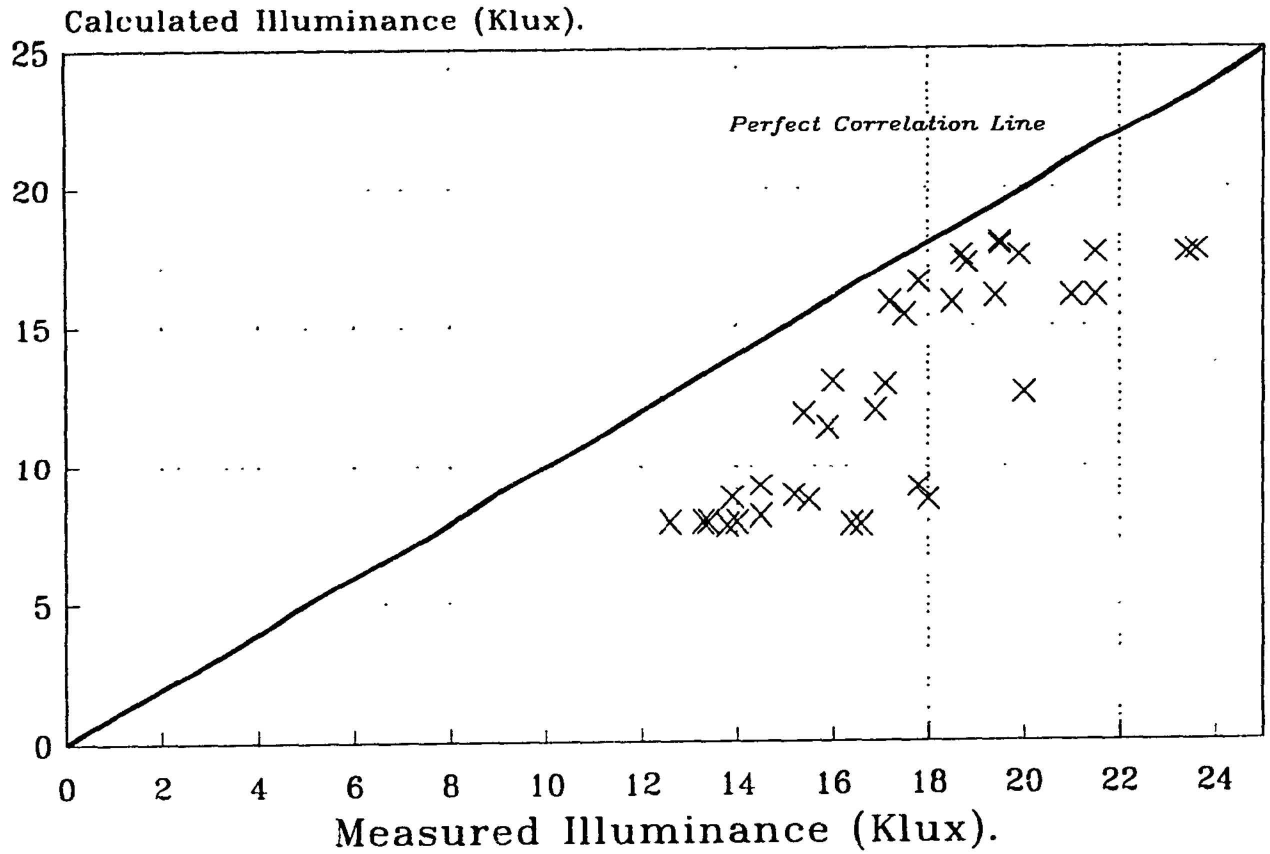


Fig 5.5 Measured Versus Calculated Illuminance on Tilted Surfaces for Overcast Sky.

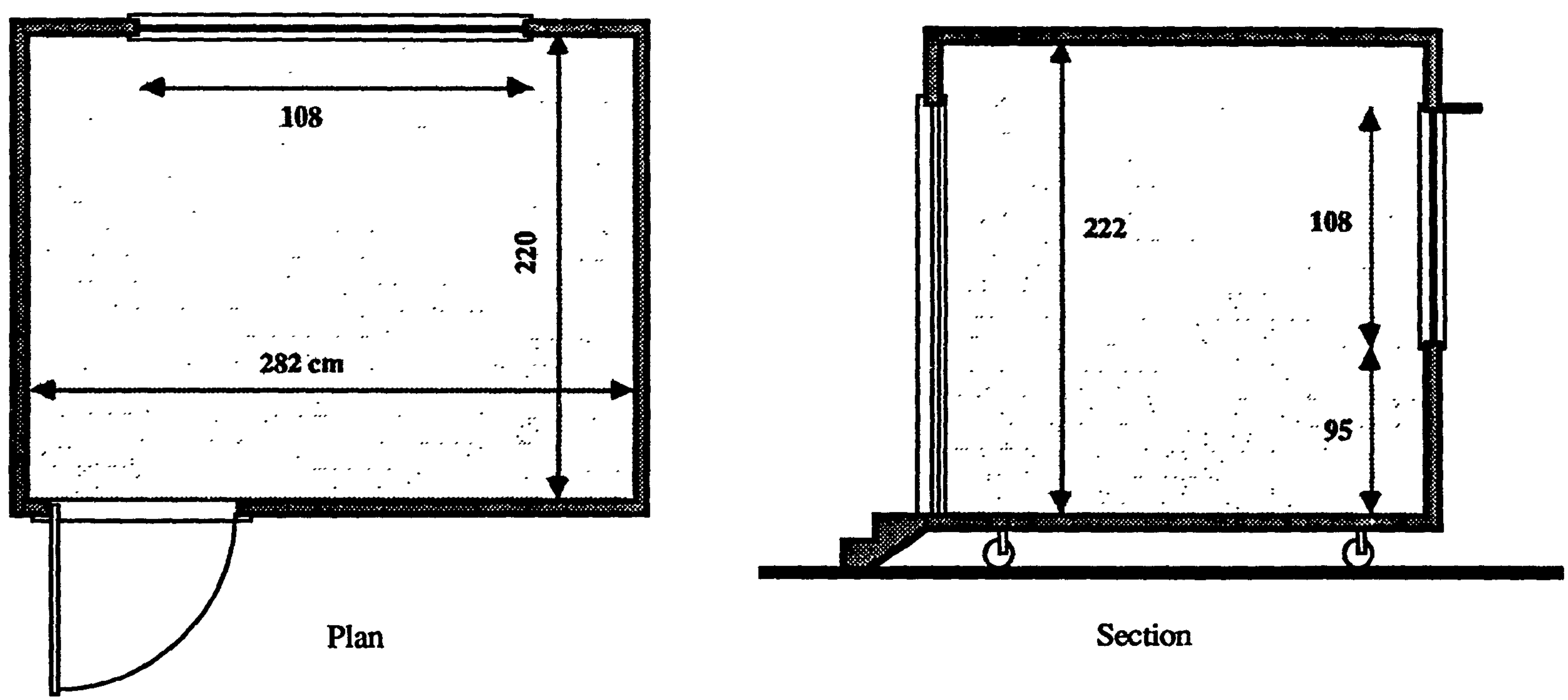


Fig 5.6 The Design of the Room Used in the Experiments.

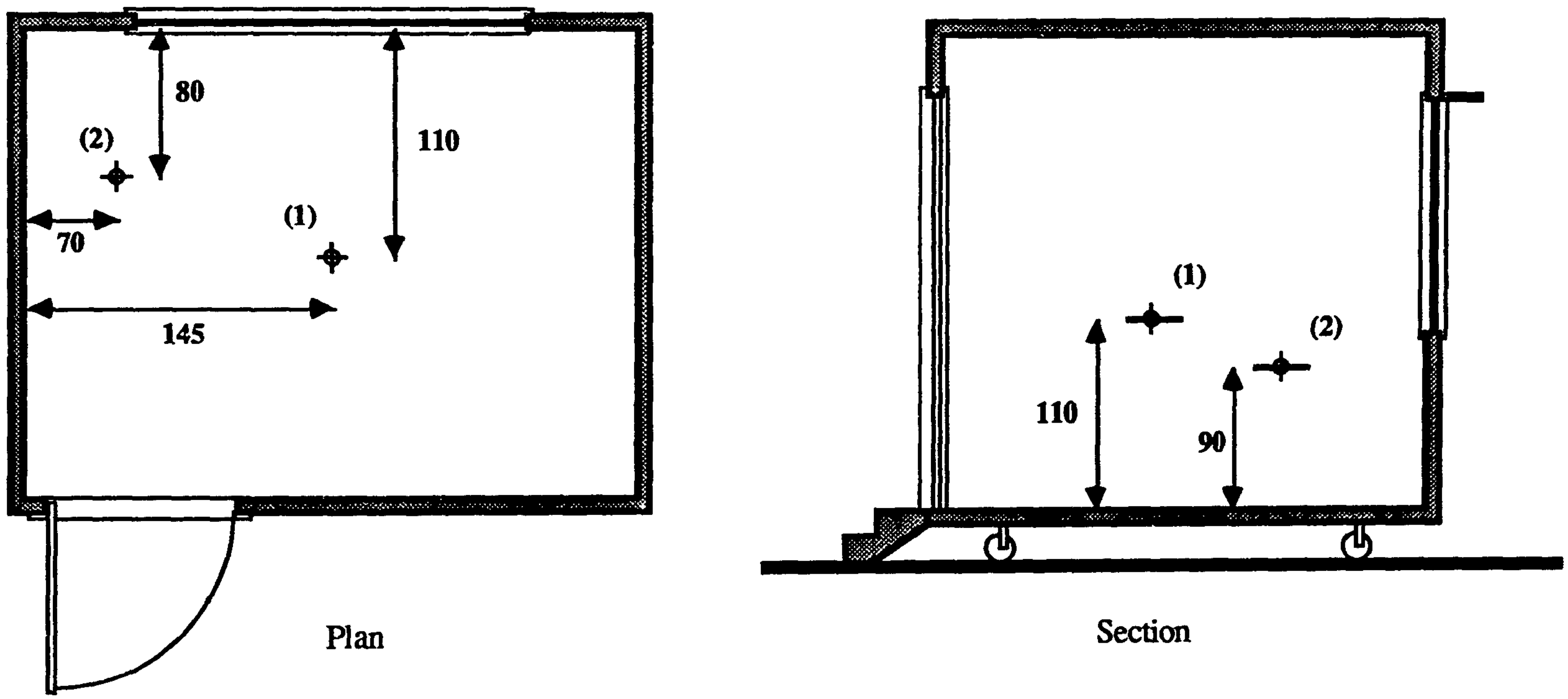


Fig 5.7 Positions of the Light Sensors in the Experiment.

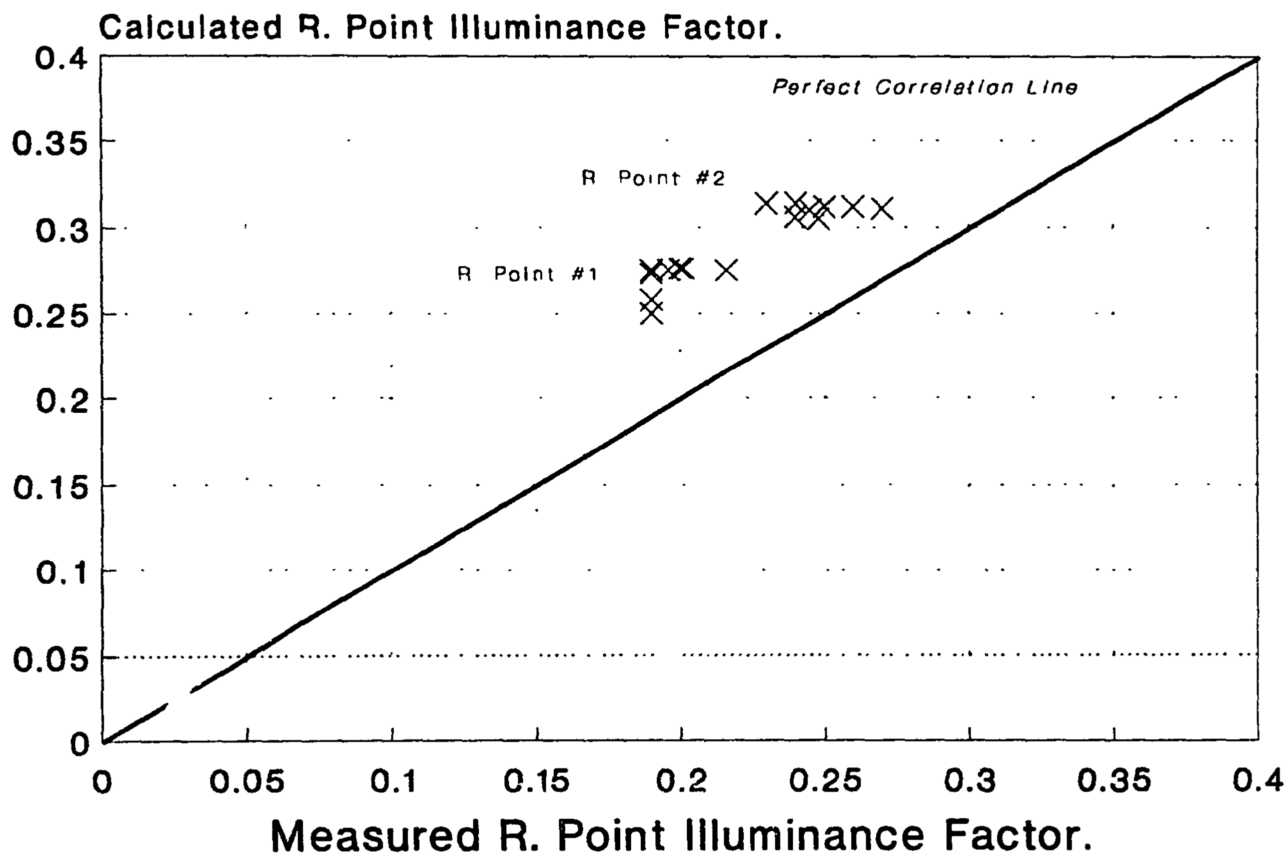


Fig 5.8 Measured Versus Calculated Reference Point Illuminance Factor.
(Illuminance on the Reference Point / Window Luminance)

CHAPTER VI

CASE STUDY AND SENSITIVITY ANALYSIS

6.1. DESCRIPTION AND AIMS

This case study was made in order to evaluate the potential for energy saving when daylight is taken into account in a passive solar building design. An abstract design is assumed to represent the general characteristics of a passive solar building. As shown in Fig. 6.1, the design is 24 x 24 m^2 in plan and 8 m in height and assumed to be a commercial building. It has one main large south facing room 12 x 18 x 8 m^3 (similar to the one shown in the application example in sec. 4.7) which is surrounded by several rooms in two stories. The large room acts as the solar collector and the thermal storage that provide heat to the rest of the building's rooms. The south window is 10 m wide and 6 m in height. It has an overhang with the same window width and projected 4.5m perpendicular to the window (see fig 6.1). The overhang is designed to eliminate the summer sun penetration to the building. The building is located in Montreal and it is assumed to have the following characteristics:

- Outside walls and roof conductance value = 0.5 $W/m^2.°c$
- Insulating Double glazed window conducıance value = 2.4 $W/m^2.°c$
- Inside wall reflectance = 0.5
- Ceiling reflectance = 0.5
- Floor reflectance = 0.3
- Window average glass visible transmittance (double glazing) = 0.65

- Area of exterior walls other than the wall of the south room = 532 m^2
- Area of windows other than the south facing window = 140 m^2
- Reference point height above floor = 0.9 m.
- Lamp efficacy = 60 Lumen/Watt.
- Luminaire coefficient of utilization = 0.45.
- Percent of lighting heat goes to plenum = 0.5
- The required illumination level is 500 lux.

An On/Off light control system is utilized for switching off the artificial lighting when the available daylight exceeds the required illumination level at any reference point. The south window is to be covered by insulating material during the night to reduce the heat loss.

This chapter aims to examine whether considering the daylight energy savings potential in direct gain passive solar buildings will have a valuable impact on reducing these buildings energy consumption. It also studies whether the building fenestration design will differ when both solar energy and daylight are taken into account in the design process. An annual analysis of the saving in the proposed building will be done followed by a sensitivity analysis for the effect of changing the south window tilt angle, azimuth angle, and area on the energy saving in the building.

6.2. ANNUAL ENERGY SAVING ANALYSIS

As discussed in section 3.5, the energy saving in the proposed building will be the difference in energy consumption between that building and its reference building. To calculate the consumption in the reference building, both heat loss from its envelope and internal heat gain from its electric lighting were calculated and plotted in Fig. 6.2. The figure shows that the internal heat gain from light exceeds the heating requirement for the reference building in some months and as a result; the reference building may consumes energy to be cooled down during these months. The auxiliary energy required to heat and cool the reference building, as well as, the energy required for illumination are plotted in Fig. 6.3. The figure also contains the total energy consumed by the reference building which is the summation of its auxiliary energy and its illumination energy.

Calculating the energy consumption in the actual building will follow the same sequence, yet both daylight and solar energy will be present in the calculation. Fig. 6.4 demonstrates the actual building heat loss through its envelop which is more than the reference building as the solar collecting window and its wall are added to the calculation. The internal heat gain from the artificial lighting is variable and less than that of the reference building as daylight utilization will lead to dimming some of the artificial light for some time. The figure also shows the solar energy affecting the building. The solar energy is higher in winter months than in summer months because the overhang prevents major portion of the direct component of the solar energy to penetrate to the building in the summer and allows only the diffused and reflected components. By adding the internal heat gain from lighting to the solar energy affecting the building, the results will be the total energy heating the building which will affect the auxiliary energy required for both heating and cooling the building as shown in Fig. 6.5. The figure also shows the energy required to illuminate the building when daylight does not provide enough illumination. The building total energy consumption is the sum of its auxiliary energy and the illumination energy. The difference between the building total energy consumption and the reference building energy consumption will be the total saving as shown in Fig. 6.6.

To appreciate the significance of considering the daylight in the calculation, the same procedures will be taken to calculate the saving in the building but when solar energy alone is considered. That simulates the conventional way of designing a direct gain passive solar building. Fig. 6.7 shows the building heat loss through its envelope and also shows the heat gain to the building by illumination and solar energy. It can be noticed that the available heat inside the building will be more in this case. That will reduce the auxiliary required energy for heating but will increase the required energy for cooling. Nevertheless, the total energy consumption will be more in this case than the case considering daylighting. That is because of the increase in the energy required for illumination (see Fig. 6.8). By comparing the total energy consumption in the reference building with the actual building total energy consumption when daylight is ignored, the saving in energy can be calculated as shown in Fig. 6.9.

Fig. 6.10 is focusing on the saved energy in the case of combining daylight with solar energy and compares it with the case of solar energy alone. The graph demonstrates the increase in energy saving in the first case during the cold months and the reduction in lost energy for cooling during the hot months. Dimming the light will result in saving the energy consumed for illumination and also reduction in the cooling load during the hot months. While in cold months, dimming the light will reduces the illumination energy, but will partially increase the auxiliary energy required to heat the building. That explains the increase in the difference between the two saving curves during the summer. By summing up the monthly saving for both curves to get an annual saving, the result will show that the saving with daylight is almost double the saving without daylight. Yet, this ratio changes by increasing the available solar energy to the building through enlarging the solar collecting window area. Fig. 6.11 and Fig. 6.12 compare the saving ratio as the window area increases from the proposed design window area until the window covers all the south wall area. This result shows that ignoring the calculation of daylight can lead to under estimation of the benefits of using the direct gain passive solar system in building design.

6.3. WINDOW TILT ANGLE

When choosing the tilt angle of the solar collecting window, it desirable to achieve more solar gain in winter and to prevent solar penetration in summer. That will occurs when the window tilt angle is close to vertical (see Fig. 6.13). Meanwhile a daylight designer will achieve better illumination distribution and higher illumination level in deep spaces when using top windows (skylights) [48]. It is expected that combining daylight with solar energy will result in a change in the optimum tilt angle for the solar collecting window in the direct gain passive solar building. Fig. 6.14 shows the results of calculating the saved energy while changing the window tilt angle for both cases; when daylight in considered and when it is ignored. The angle of maximum saving has been shifted as indicated on the figure.

6.4. WINDOW AZIMUTH ANGLE

Facing south allows the solar collecting window to receive solar energy for the longest period of time, which is required during the heating season. But to avoid overheating in summer, the window is usually provided with an overhang that prevents the high altitude summer sun from penetrating the building and increases the cooling load. As the overhanged solar collecting window rotates towards east or west, it faces lower altitude sun during longer period of time. This low altitude sun penetrates the building and increases the available illumination in it. It also increases the cooling load in the building during the summer. The balance between increasing the illumination and increasing the cooling load determines the optimum window orientation during summer time and consequently throughout the year. When daylight is ignored, only the solar energy effect on the cooling load will affect the determination of the optimum azimuth angle.

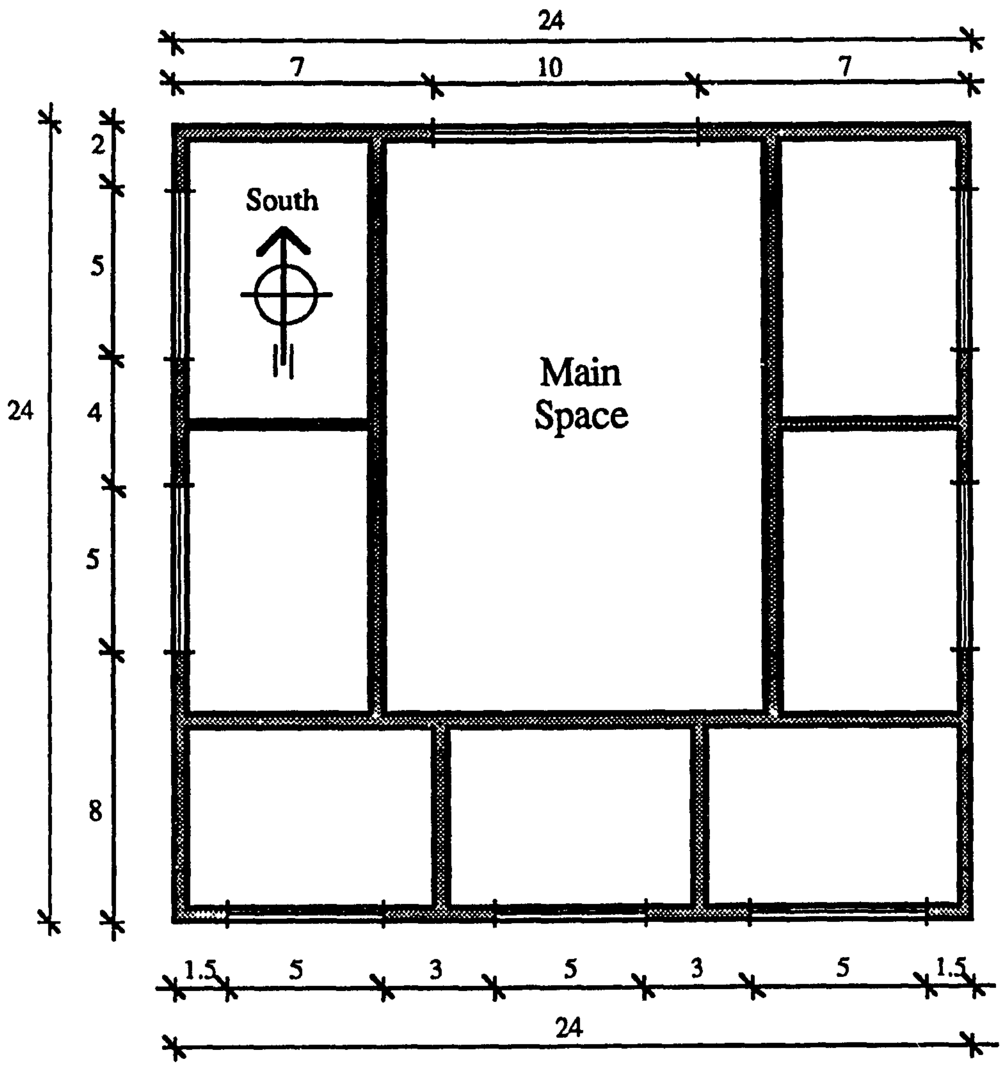
Fig. 6.15 shows the annual energy saving in the proposed building as a function of changing the azimuth angle of its solar collecting window. The figure indicates that the maximum saving occurs when facing south for both cases. The large window provides enough daylight to the space throughout the year even when the solar spot is prevented by the overhang in summer time. But when the solar collecting window area is reduced to half the initial area, the solar spot has more effect on the illuminance level in the deep areas in the space. Fig. 6.16 shows that shifting from south increases the annual energy saving in the building in that case. It also shows that the azimuth angle for the maximum saving has changed in the curve represents utilizing daylight from that represents ignoring daylight. The balance between saving in lighting and increasing the cooling load makes that change.

6.5. WINDOW AREA

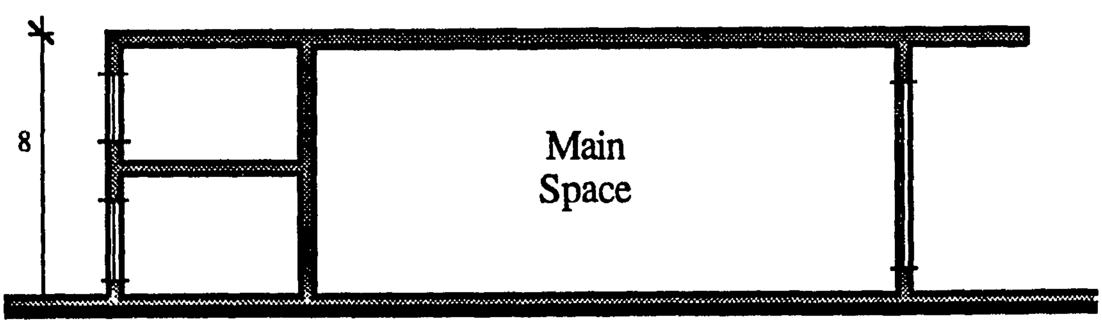
To examine the impact of the window area and whether there might be an optimum area for increasing the potential energy saving, the south window ratio to its wall was changed from 2.5% to 100% with an increment of 2.5%. To keep the same effect on sun penetration in all the cases, the overhang width and projection were adjusted. Fig. 6.17 displays the results for the proposed building. The figure shows that the more window area available, the more the saving will be in both utilizing and ignoring daylight. That is due to the high energy requirement to heat the building in Montreal. Yet, if the same building is studied in warmer climate, a very large glazing area will result in high energy consumption to cool down the building in summer. Fig. 6.18 shows the result of locating the proposed building in Washington. The figure shows that the maximum saving in energy when combining daylight with solar energy has been achieved in a window ratio different than that achieved when solar energy was utilized alone.

6.7. DISCUSSION

The results of this case study have indicated the importance of considering daylight in the energy analysis and the design of the direct gain passive solar buildings. The daylight contribution in energy saving makes these types of buildings more attractive and more promising. The results also show that combining daylight with solar energy may have an effect on the fenestration design so as to achieve the maximum energy saving. This effect was limited in this case study, yet it can be more significant in other cases.



Plan



Section

Fig 6.1 The Proposed Abstract Design for the Case Study

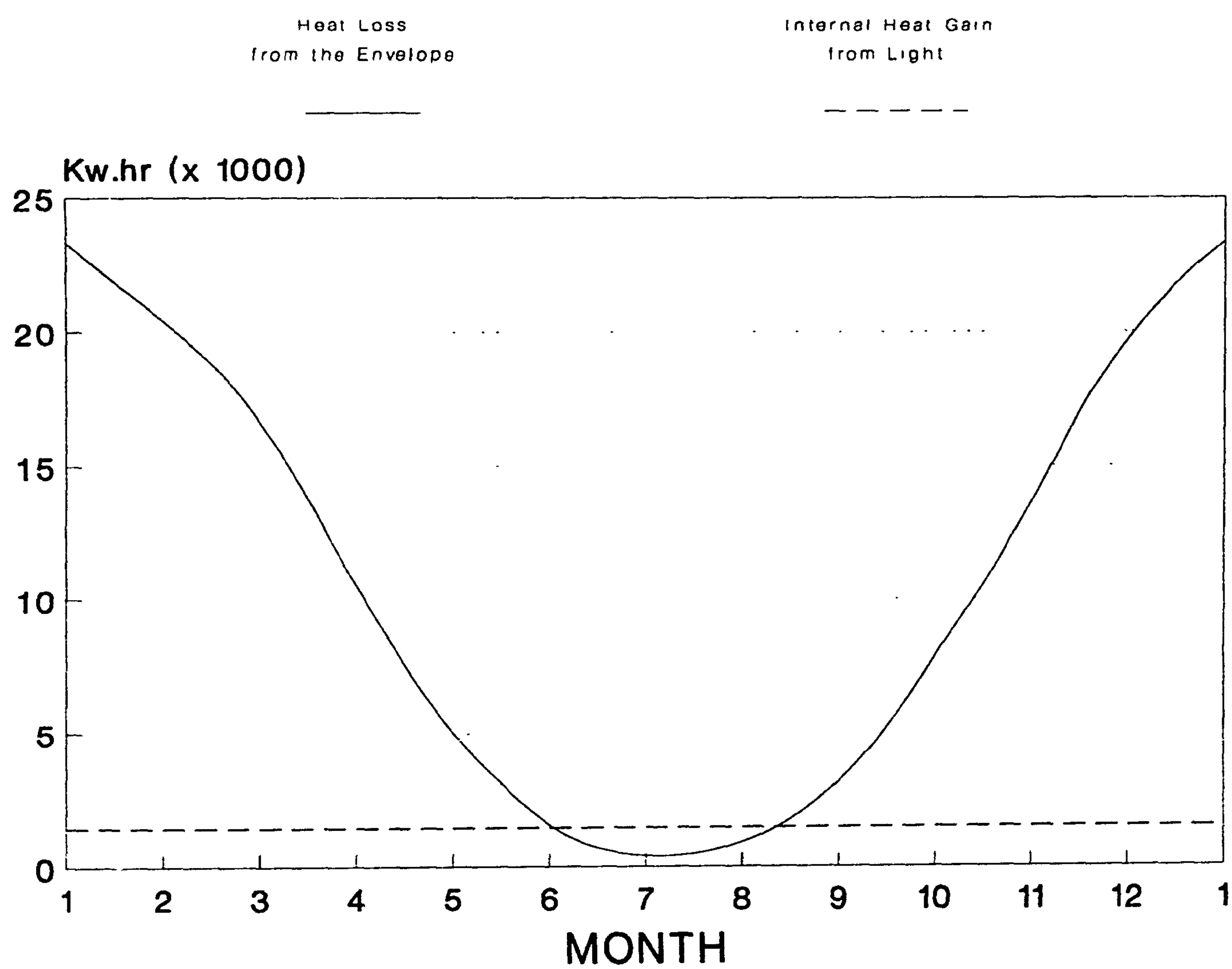


Fig 6.2 Reference Building Monthly Heat Loss through the Envelope and the Internal Heat Gain from Light

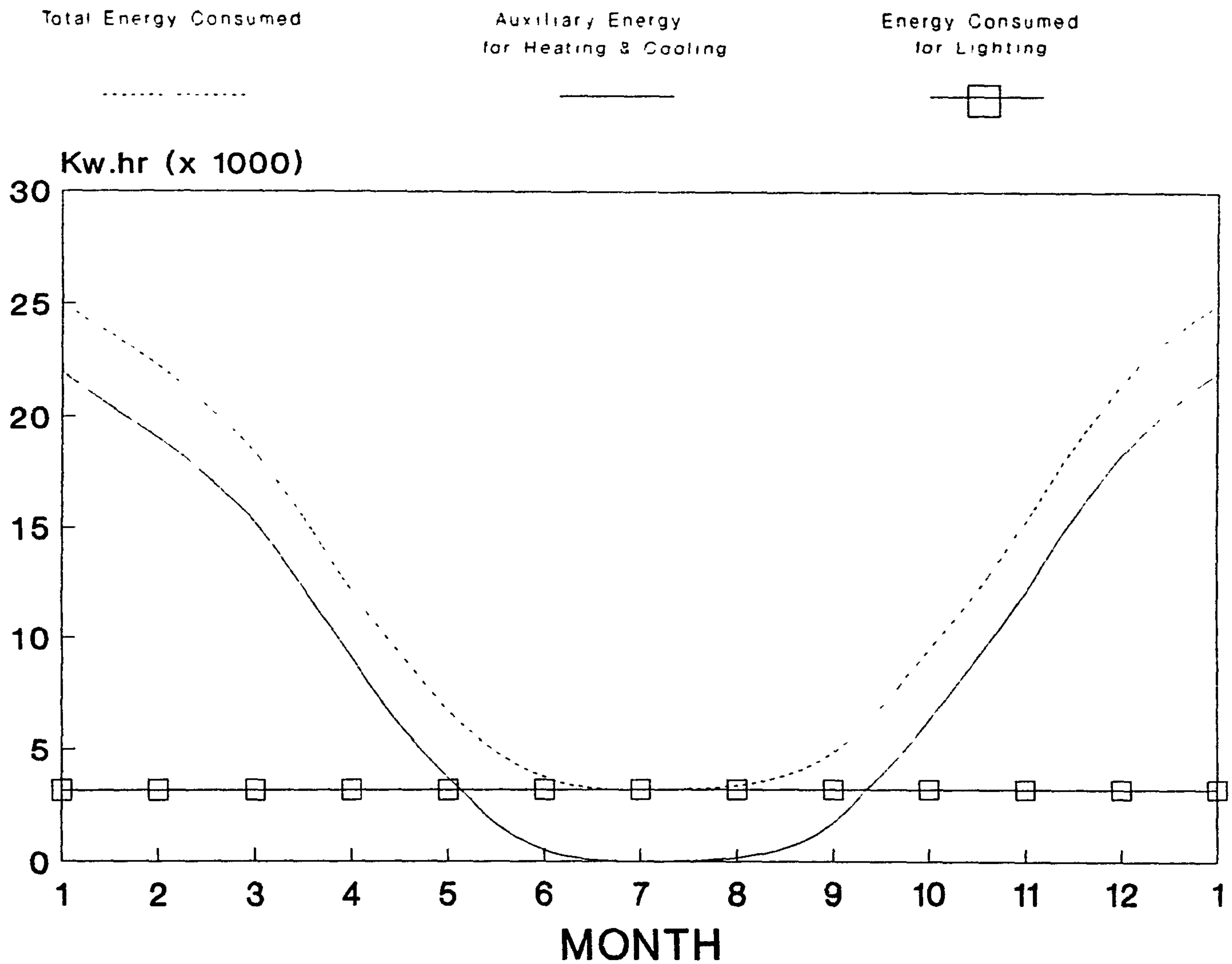


Fig 6.3 Reference Building Components for Monthly Energy consumption.

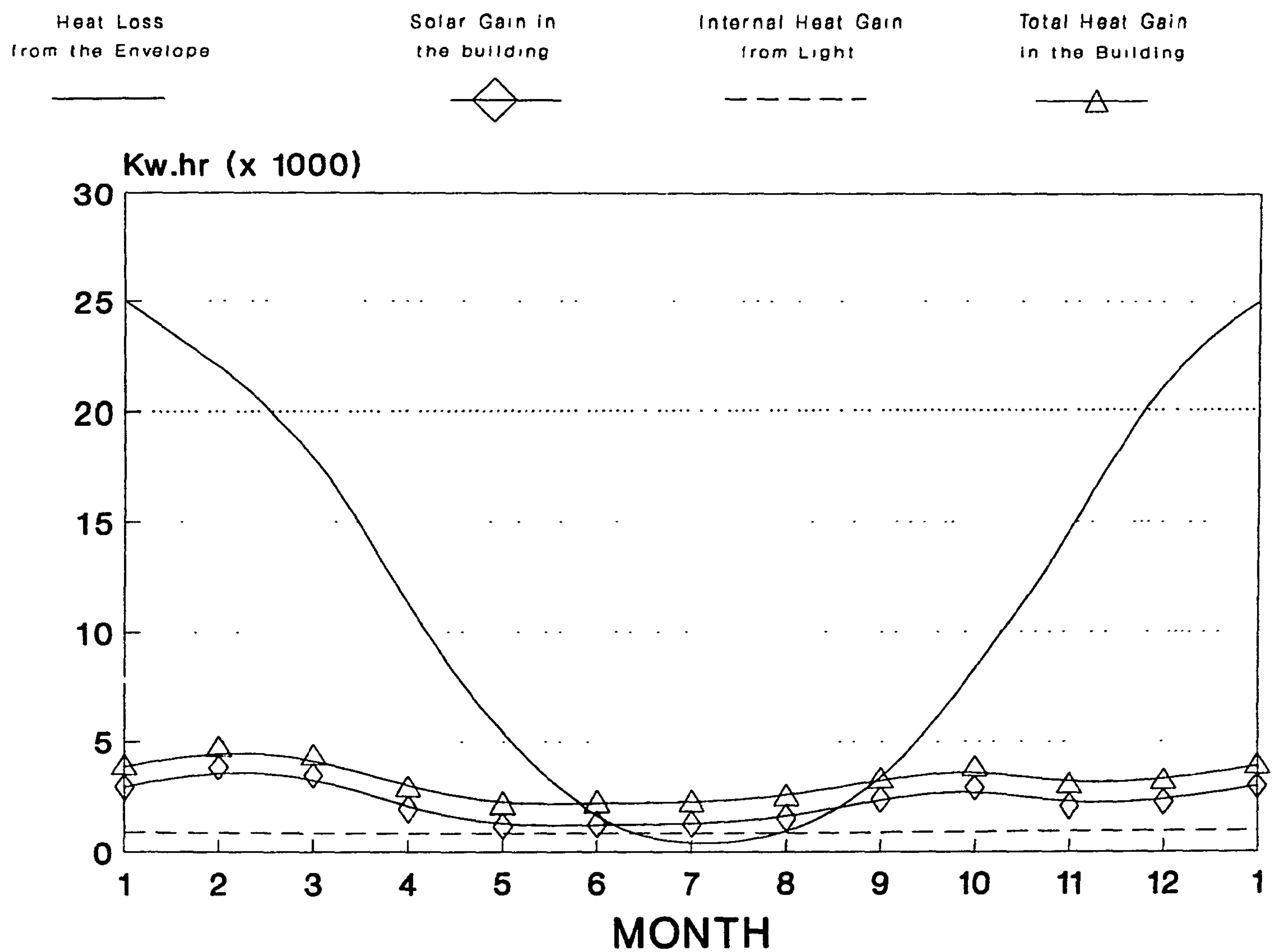


Fig 6 4 Actual Building Monthly Heat Loss througn the Envelope and the Heat Gain in the Building (Combined Solar Energy and Daylight)

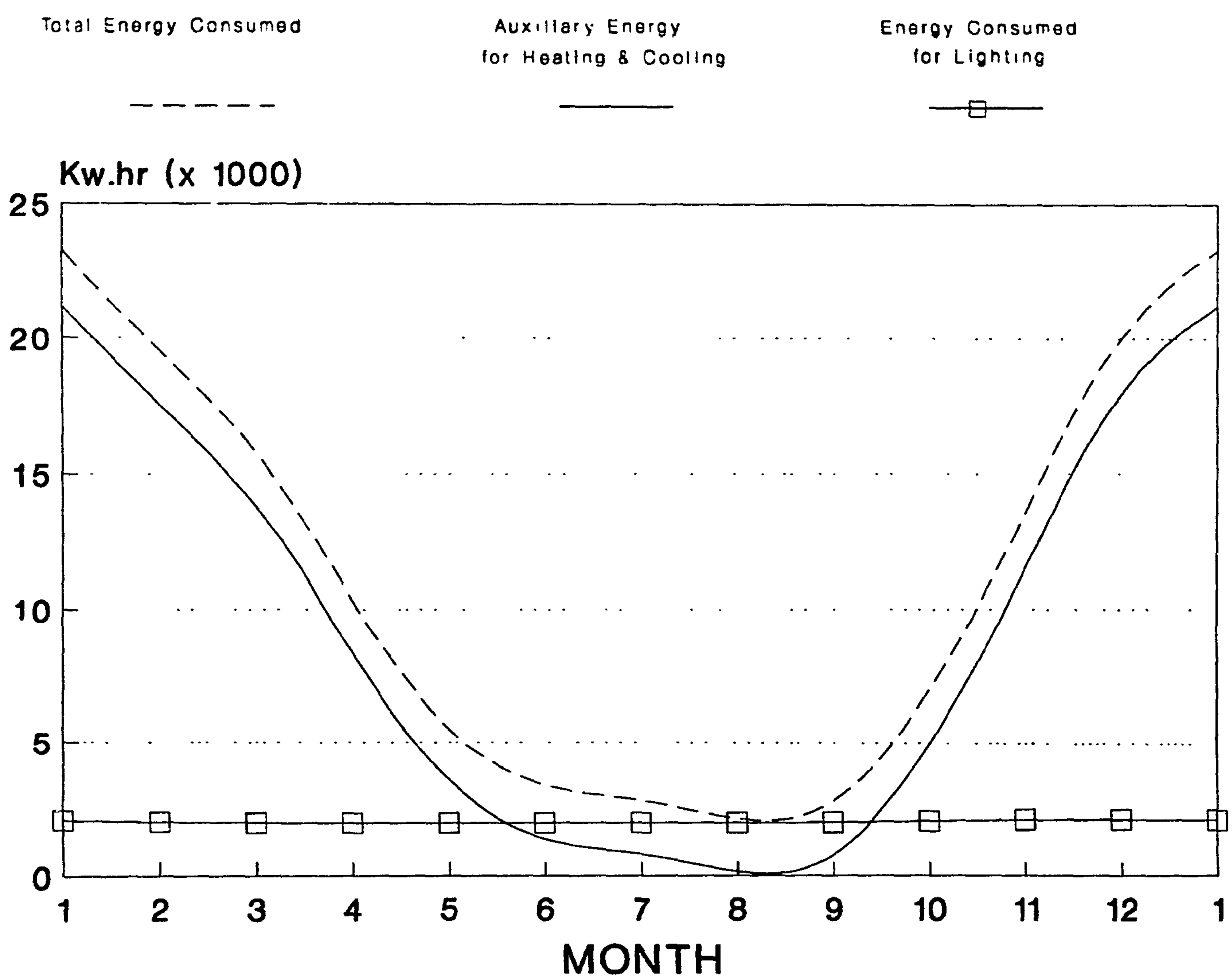


Fig 6.5 Actual Building Components for Monthly Energy Consumption (Combined Solar Energy and Daylight)

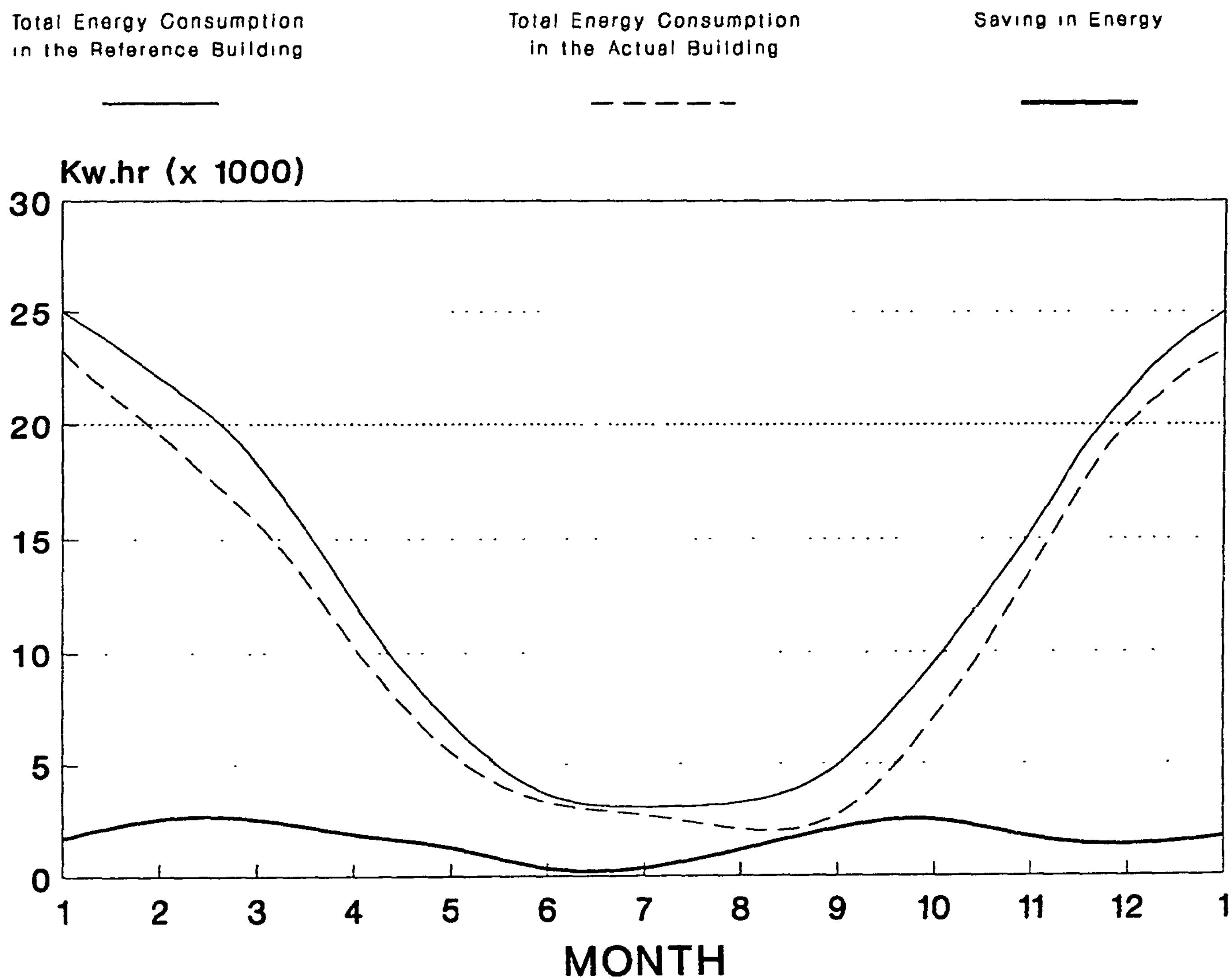


Fig 6.6 Monthly Saving in Energy Consumption
when Daylight is Combined with Solar Energy

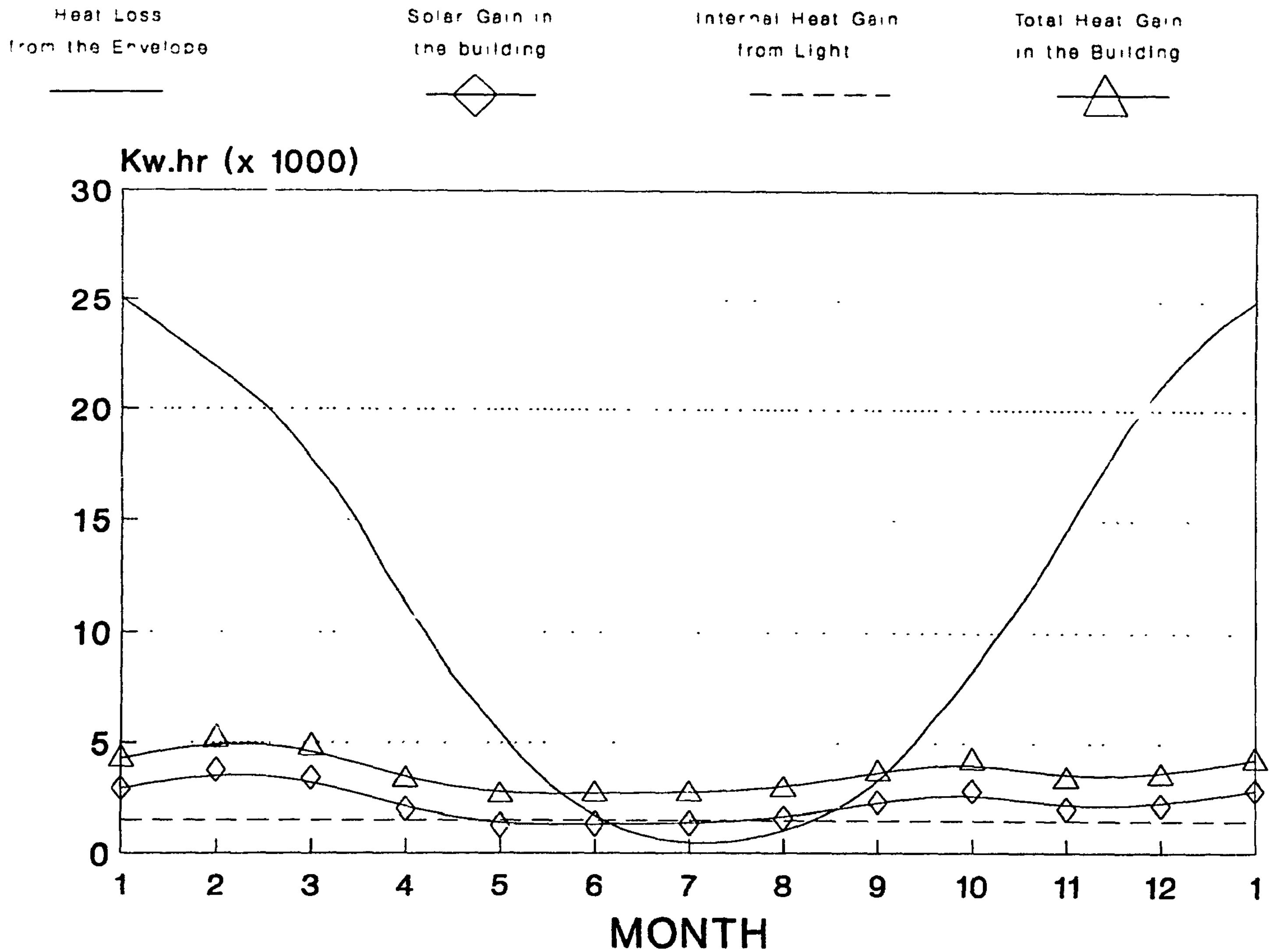


Fig 6 7 Actual Building Monthly Heat Loss through the Envelope and the Heat Gain in the Building. (Solar Energy Alone)

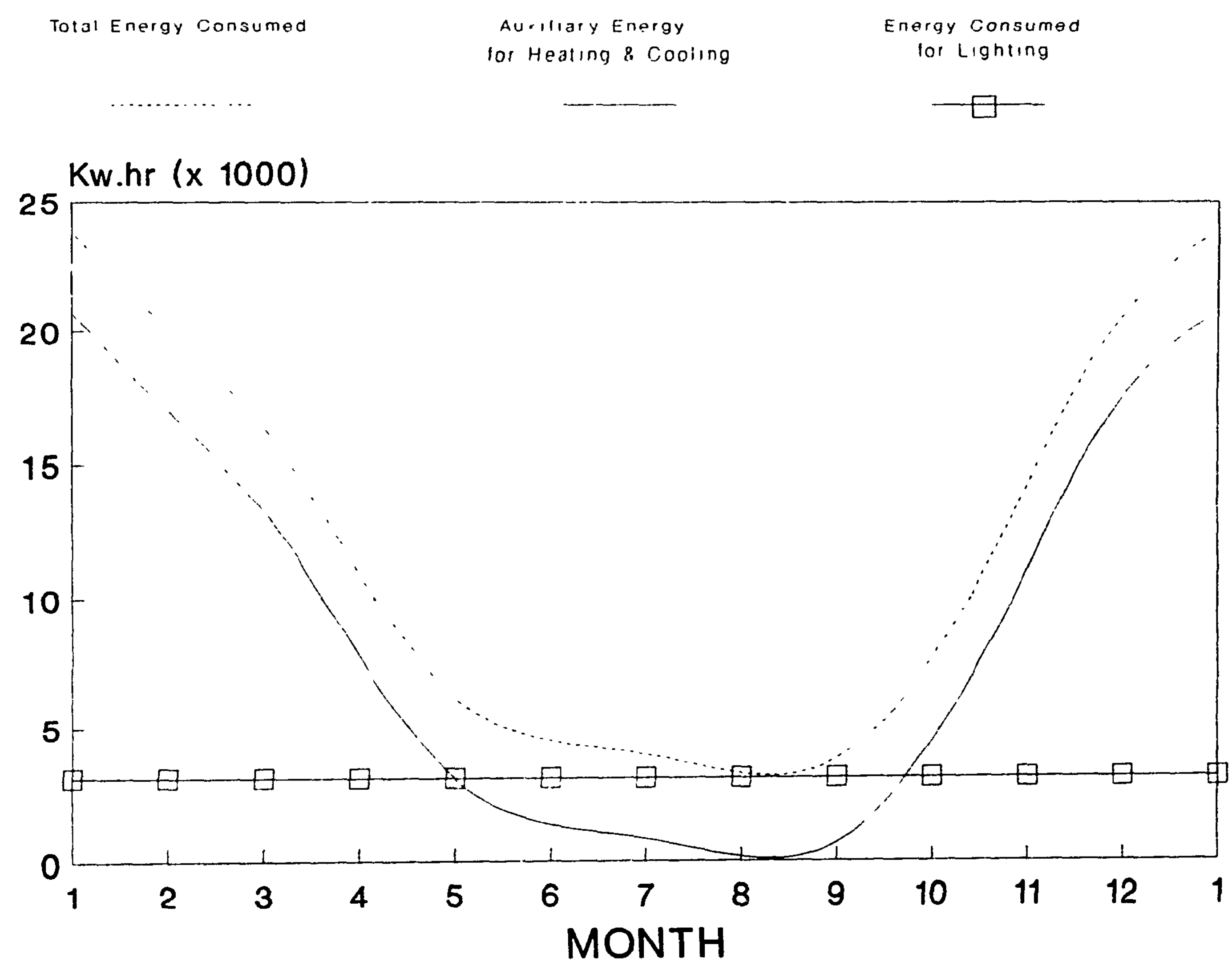


Fig 6.8 Actual Building Components for Monthly Energy Consumption (Solar Energy Alone)

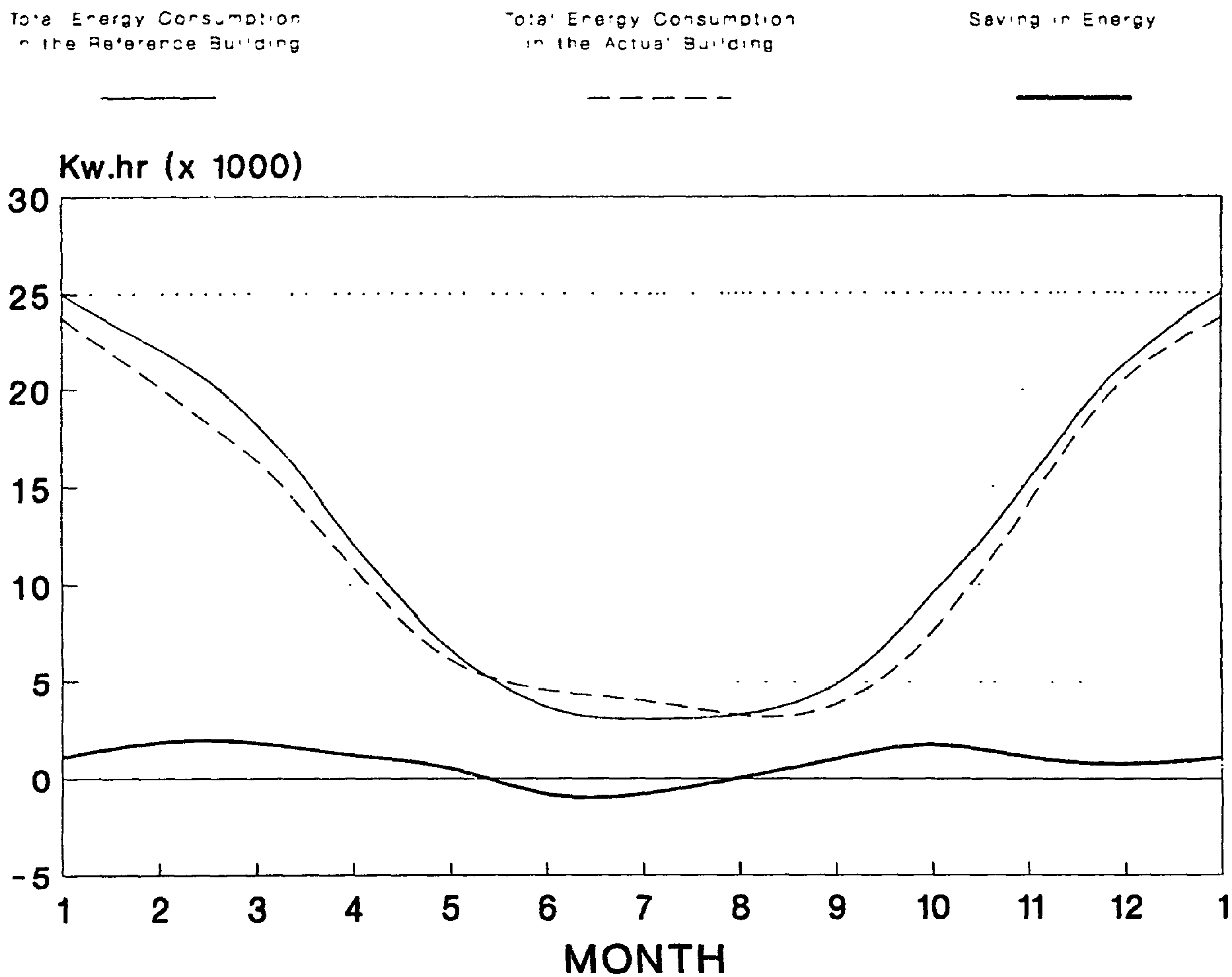


Fig 6.9 Monthly Saving in Energy Consumption when Solar Energy Alone is Considered.

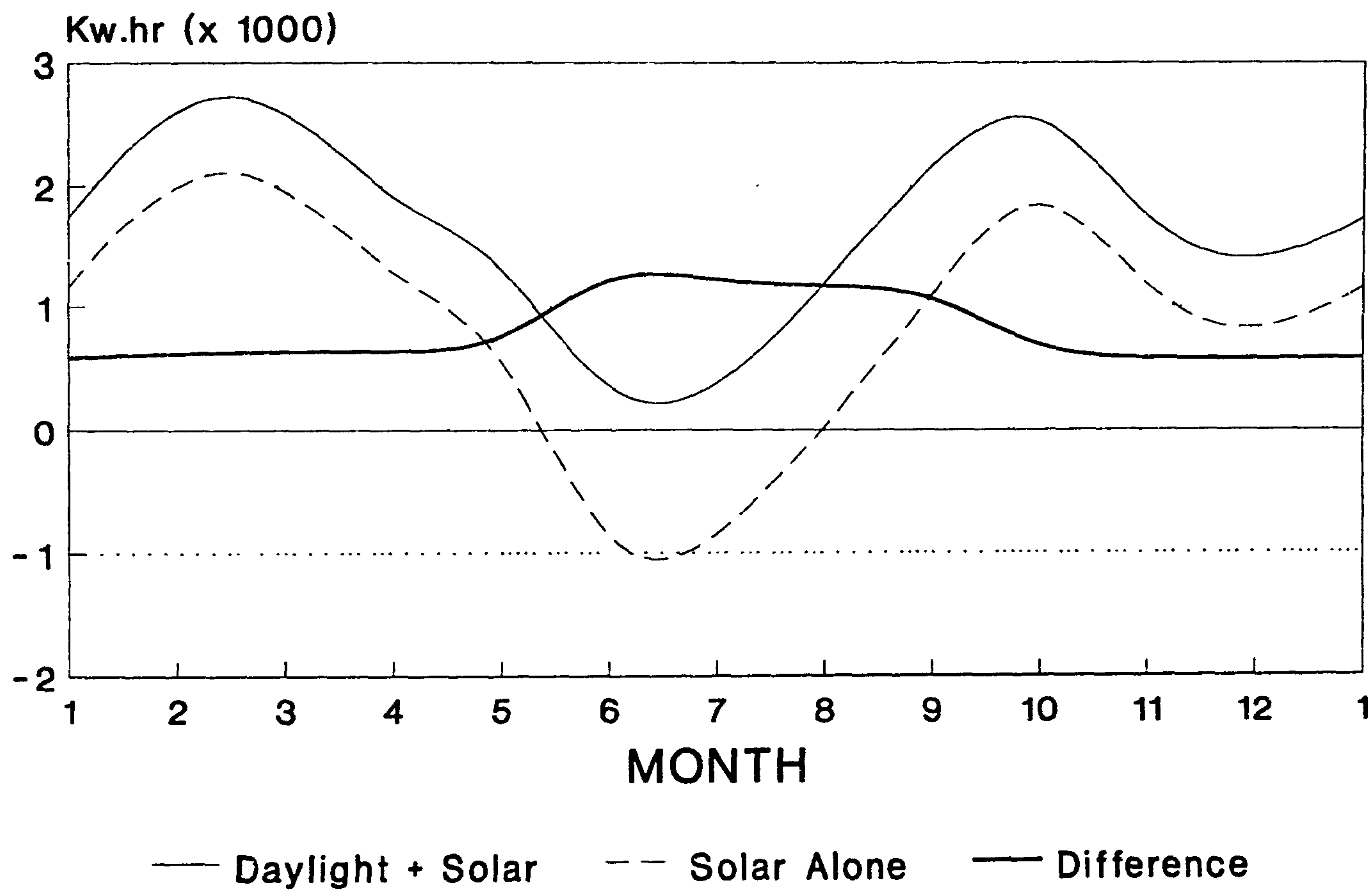


Fig 6.10 Monthly Saving in Energy Consumption

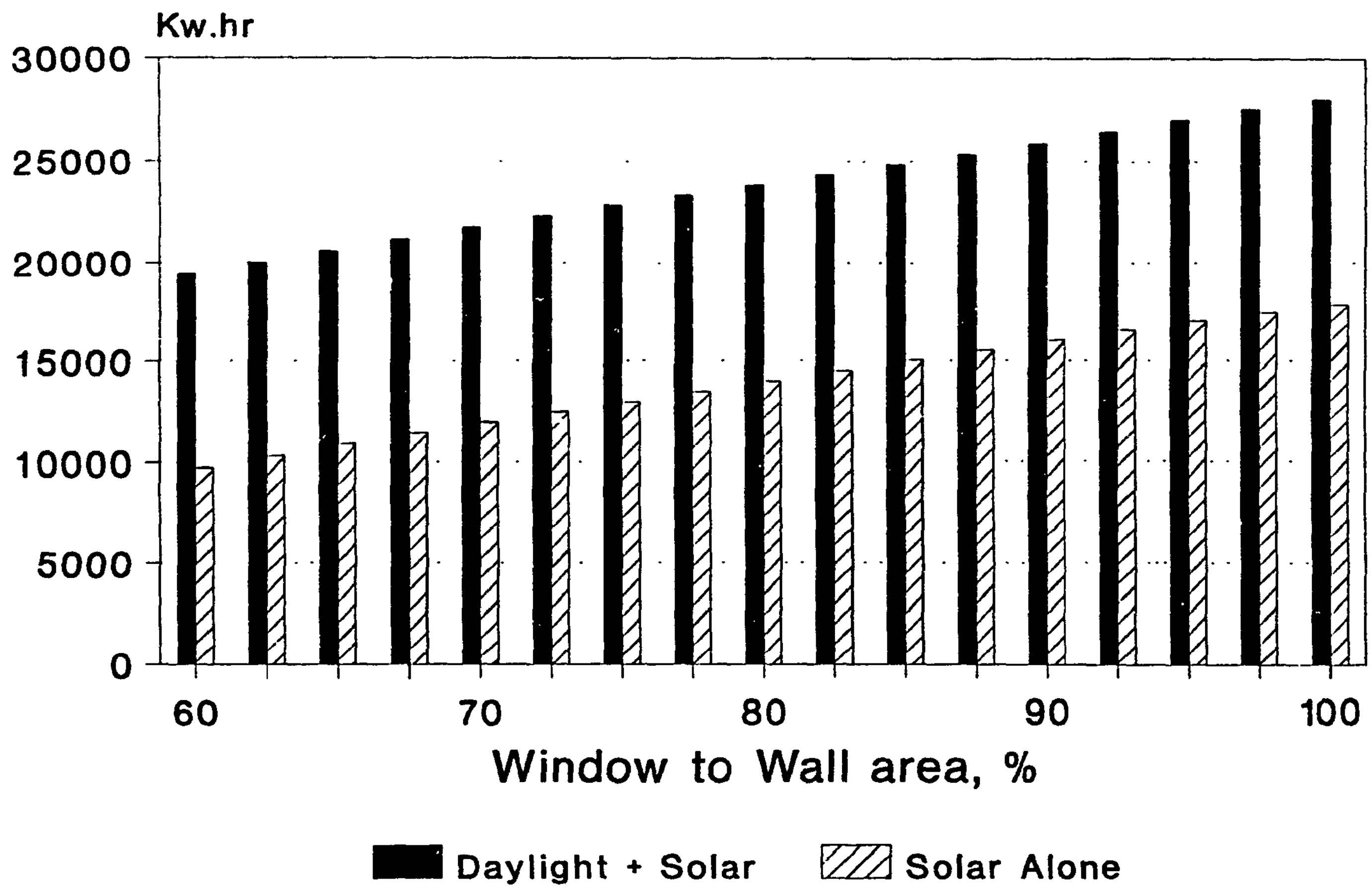


Fig 6 11 Increase in Annual Energy Saving due to Combining Daylight with Solar Energy.

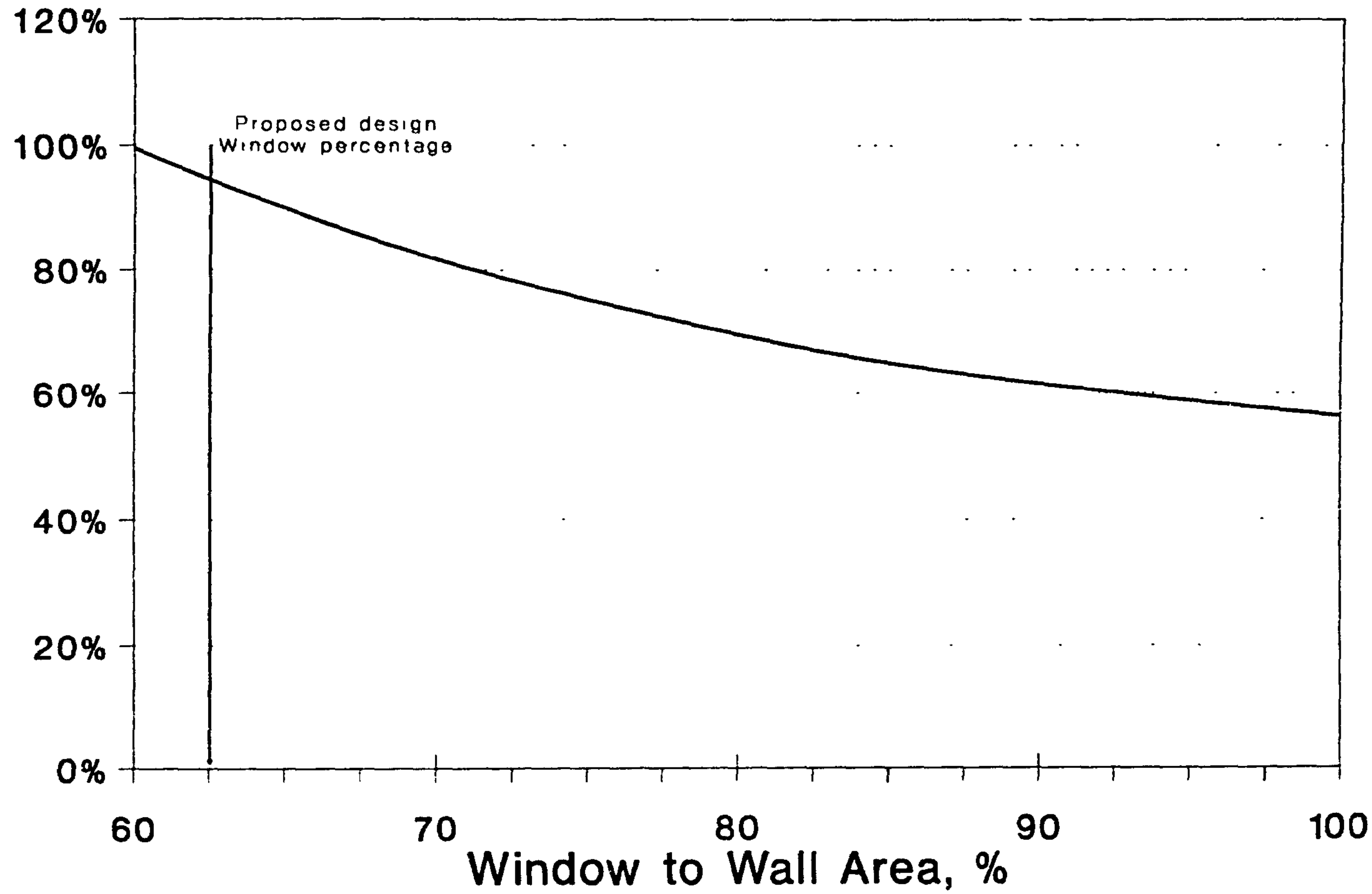


Fig 6.12 Percent Increase in Annual Energy Saving due to Combined Utilization of Daylight with Solar Energy Over the Use of Solar Energy Alone

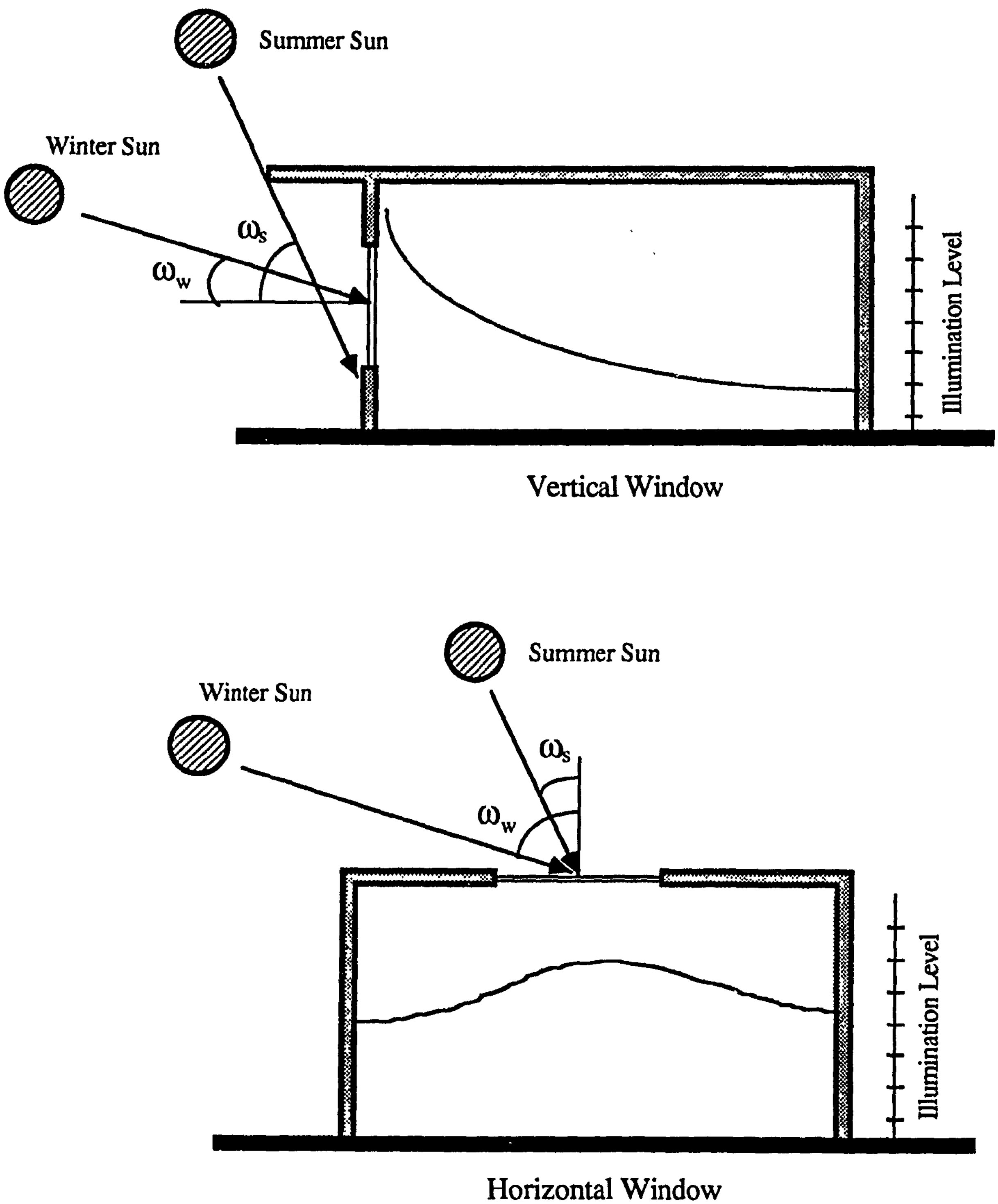


Fig 6.13 Comparizon between horizontal and vertical windows

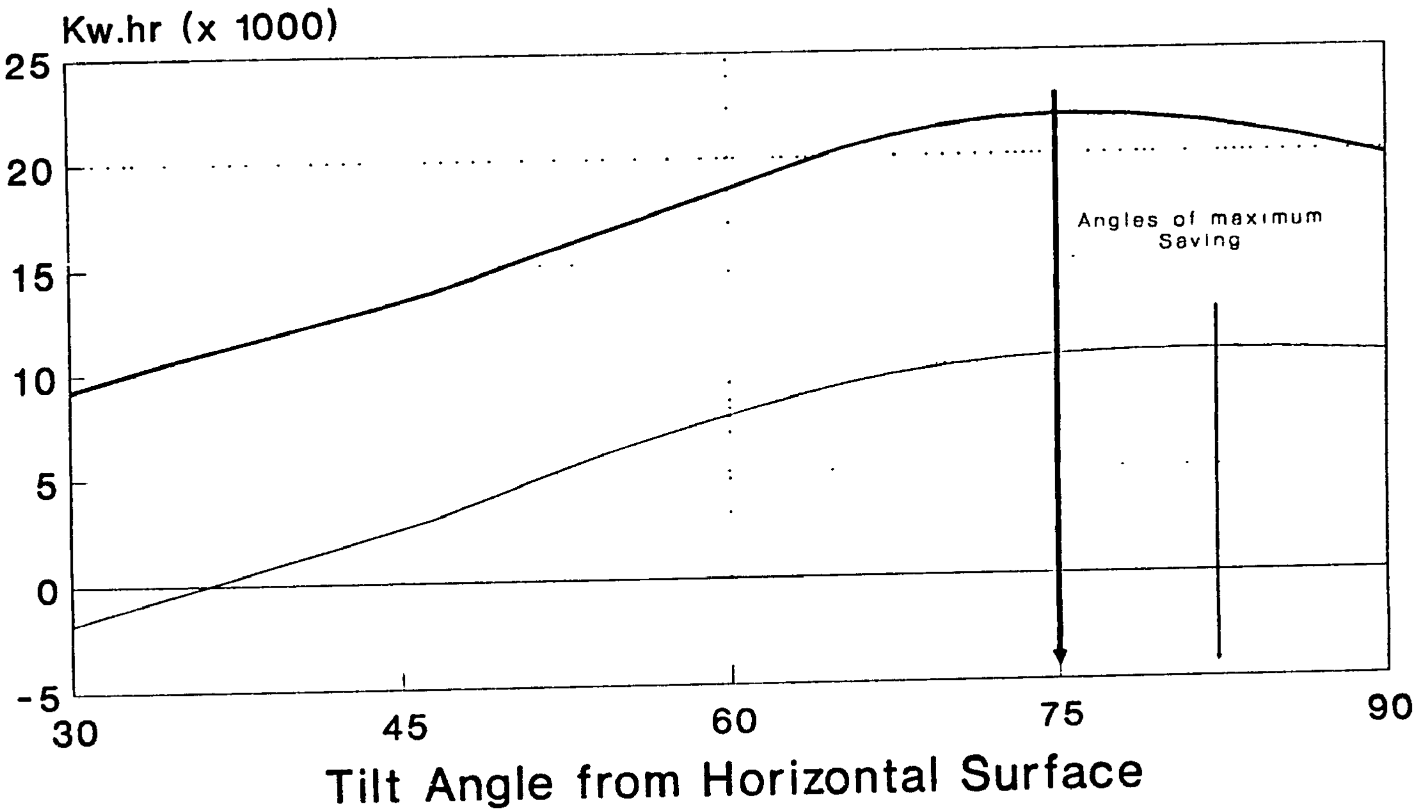


Fig 6.14 Annual Energy Saving As A Function Of Window Tilt Angle

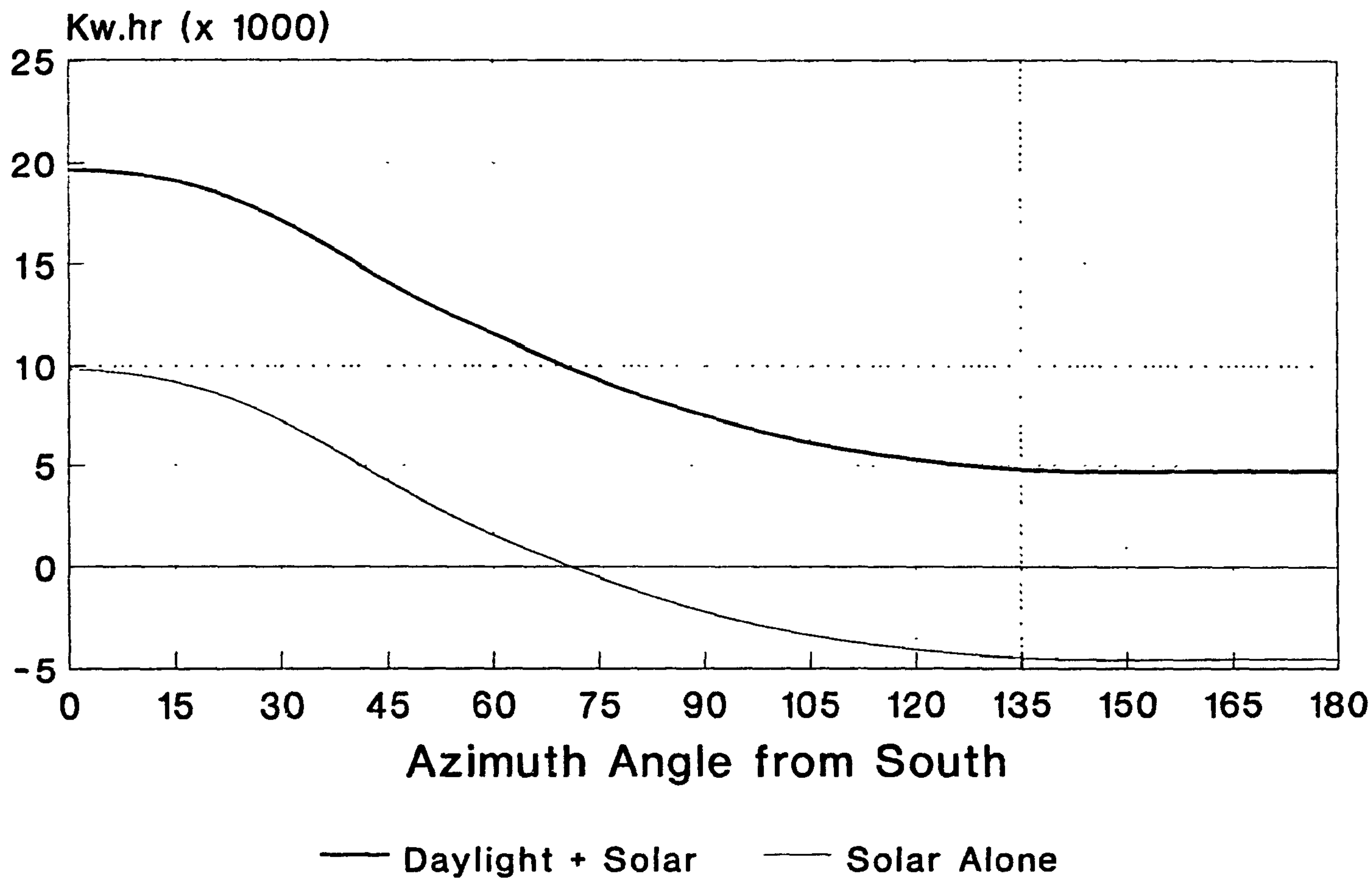


Fig 6.15 Annual Energy Saving As A Function Of Window Azimuth Angle.

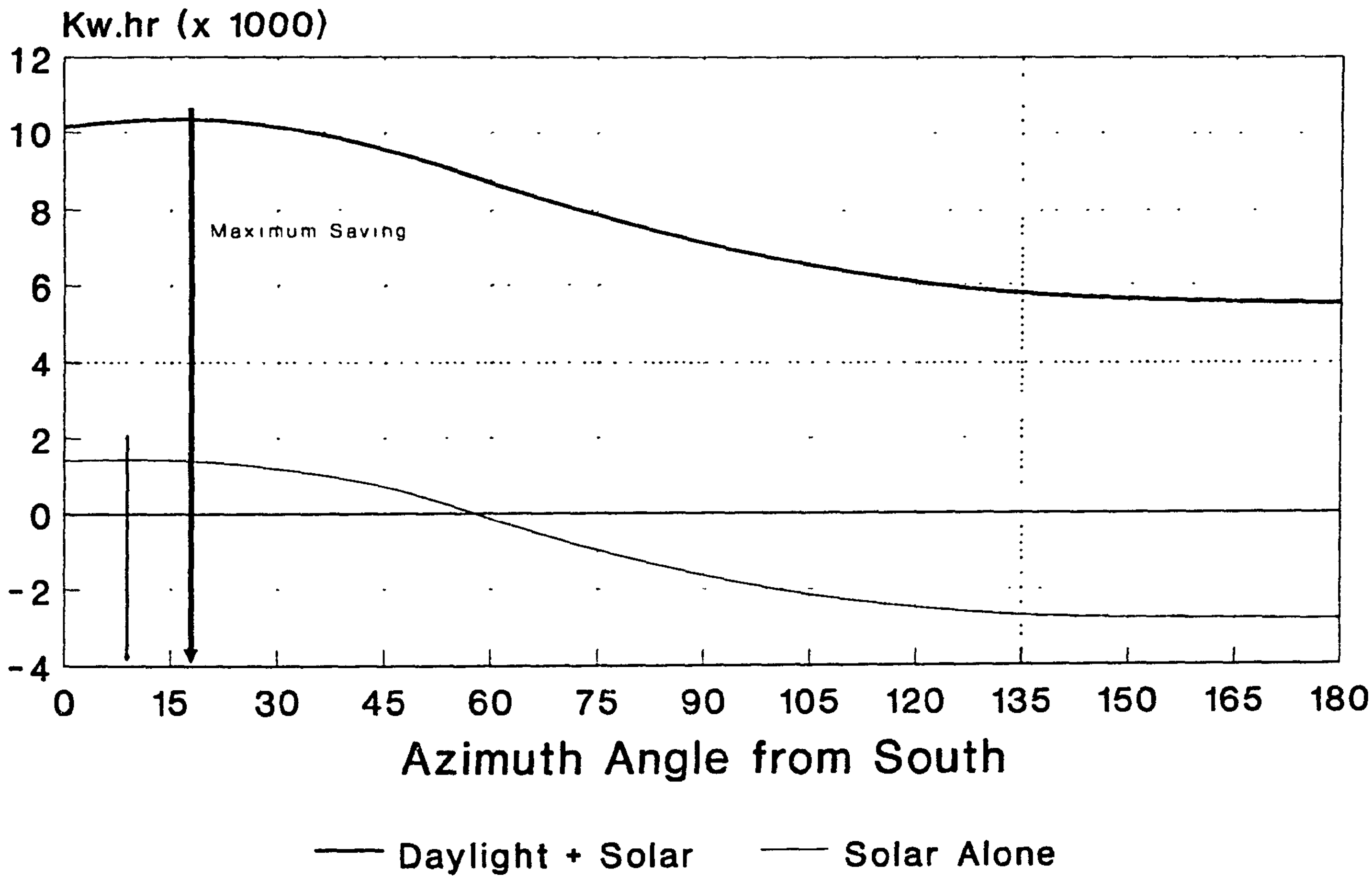


Fig 6.16 Annual Energy Saving As A Function Of Window Azimuth Angle (Window = 25% of Wall Area)

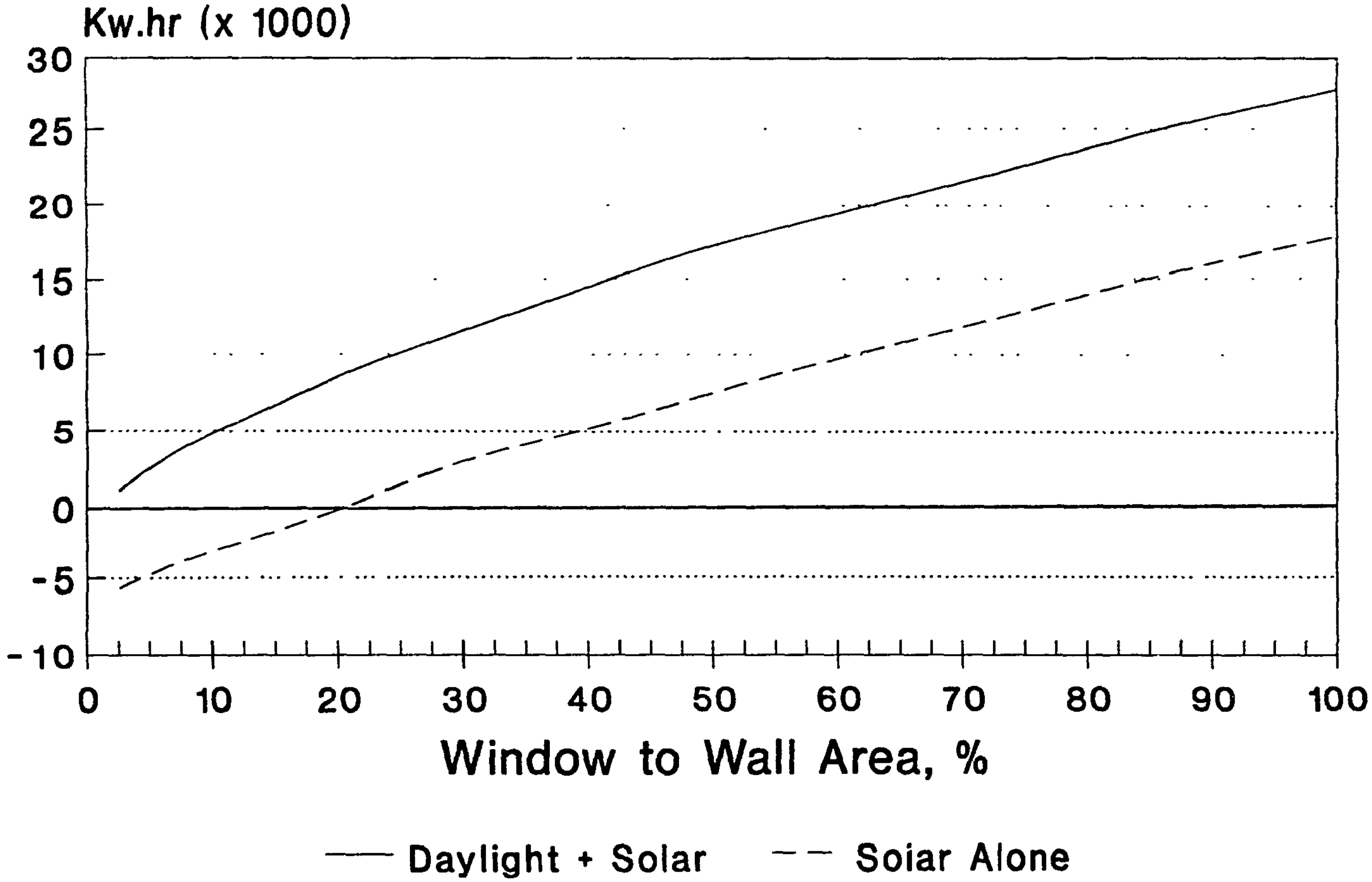


Fig 6.17 Building Annual Saving As A Function Of Window Area.

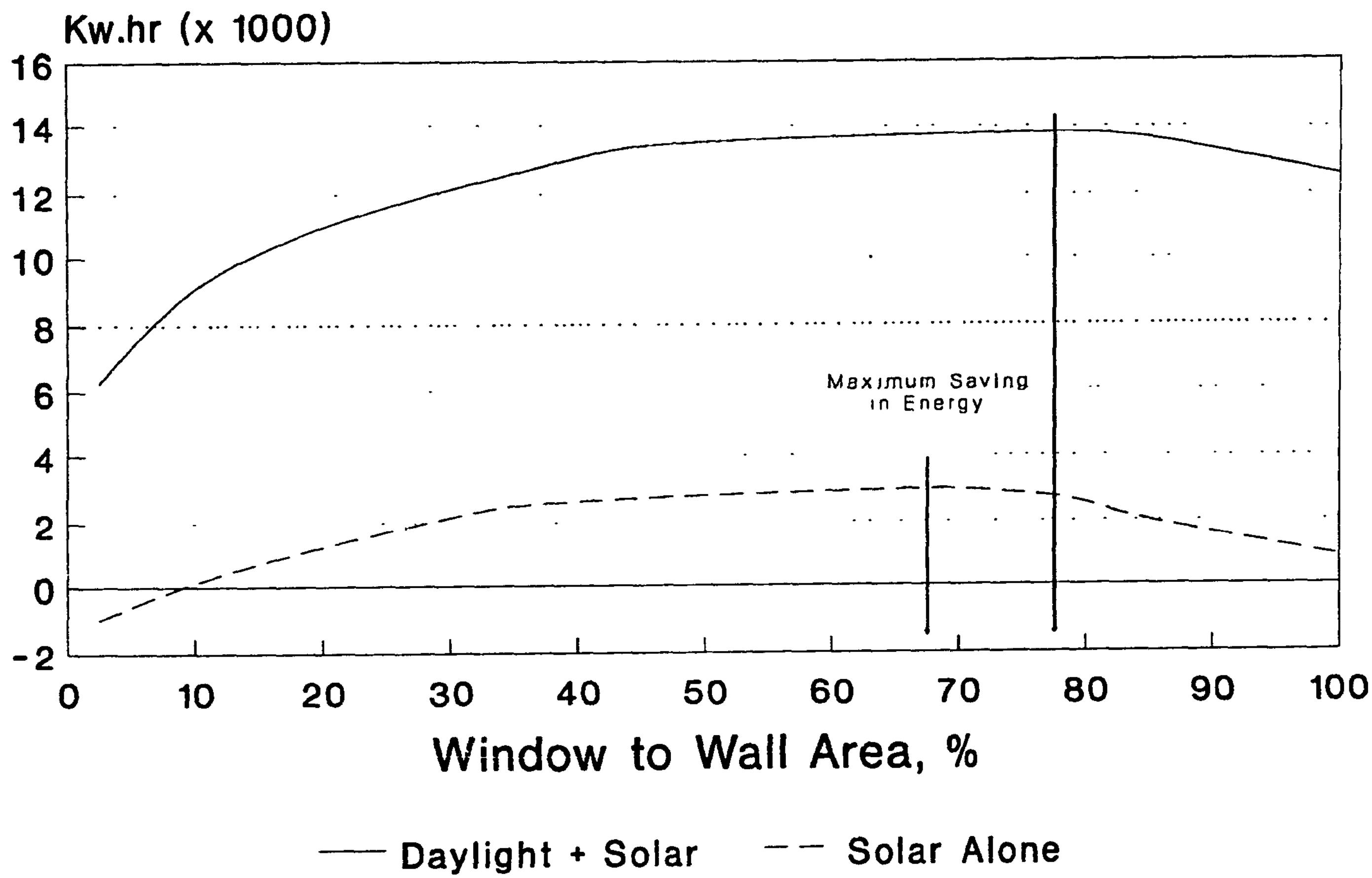


Fig 6 18 Building Annual Saving As A Function Of Window Area (in Washington)

CHAPTER VII

CONCLUSION

7.1 SUMMARY AND CONCLUSION

Direct gain passive solar buildings are characterized with large glazing area. As a result a large amount of daylight is available in the interior of these buildings. This free source of illumination is getting more attention as a concept for designing in direct gain passive solar buildings, particularly in commercial buildings where electric lighting represents a large portion of the energy consumption. Several methods of assessment have been introduced for daylight design and others for passive solar design but none attempted to incorporate both in sufficient depth. In this study, a mathematical model that incorporates both of solar gain and daylight has been developed to assess the combined saving potential. The model first quantified the dynamic availability of daylight from its two components, the sun and the sky. Clear, partly cloudy, and cloudy sky conditions were considered. Illuminance can be calculated from both the sun and the sky at any time over a surface which have any azimuth and any tilt angle. The model calculates next the illuminance on any reference point in a rectangular room due to the available daylight on the room window and due to the solar spot resulting from the sun penetration through that window. An algorithm has been developed to calculate the location of the solar spot in the room at any time and consequently the illuminance effect of that spot on the room surfaces. In parallel with the natural lighting calculation, the model determines the dynamic availability of solar energy and calculates the heat gain in the room. All these information is then used in a developed methodology that estimates the saving in energy from the combined utilization of solar gain and daylight. This saving is calculated by comparing any given room with a reference room. The reference room is having the same room design but the solar collecting window and its wall are replaced by a neutral wall that does not permit any type of heat transfer . The methodology takes into account both the negative and the positive effect of dimming the artificial lighting on the building energy consumption. An hour by hour calculation for both daylight and solar gain is utilized to determine the saving in any given day. For the calculation of annual savings, the 21^{st} day of each month was assumed to represent that month and the sum of the monthly saving will result in the annual saving. This methodology assumes well designed thermal mass within the direct gain passive solar building that is able to distribute the solar gained to the building evenly through that month. So over heating will occurs only if the monthly solar gain to the building accedes its monthly heating requirements. Cooling loads are calculated using the transfer function method which utilize the hourly available information on solar gain, internal heat gain from lighting, and outside temperature as well as it consider the effect of the building material used. The methodology aims to provide an estimate for the annual energy saving and evaluate the building energy saving potential of different design alternatives. The model was partly constructed using formulae recommended by the IES and by ASHRAE. Other formulae used were evaluated using experimental work and proved to be fairly credible.

The mathematical model has been coded into a computer program. A case study was developed to determine the energy saving potentials of combining the daylight with the solar gain. The building used in the case study represents a typical design of commercial direct gain passive solar building. The program was also used to conduct a sensitivity analysis for factors influencing the window design to study their effect on the saving in energy. The results have shown that combining daylight with solar gain has almost doubled the energy saving in the proposed case study. It also indicated that the optimum window orientation, tilt angle, and area have been changed in case of combining daylight and solar gain from the case when solar gain is considered alone.

The research results advise the direct gain passive solar building's designers to study the daylight availability and distribution in their designs and to be aware of the high potential of a good daylight design in increasing their design efficiency. This study should also increase the attractiveness of utilizing the direct gain passive solar system in the small and medium size non domestic buildings.

7.2. CONTRIBUTIONS

This research has offered several contributions:

- The potential saving associated with daylighting when utilized in the direct gain passive solar buildings is explored and the solar collecting fenestration dependence on its tilt angle, orientation, and area when daylight needs to be combined with solar heat gain was demonstrated.
- A basic model for combining daylight and solar energy in the energy analysis of the direct gain passive solar buildings was developed.
- A method to predict the dynamic sunspot location in a room and its consequent

illumination impact on a point using 15 different cases was developed.

- A calculation method for energy saving due to daylight and its combination with solar energy was developed.
- A computer software that can assists designers in comparing the energy saving potentials of their design alternatives when they combine the daylight with solar energy was developed.

7.3. RECOMMENDATIONS FOR FUTURE WORK

This work can be a base for further future research in the area of utilizing daylight in passive solar buildings. The following areas are worth consideration in future studies:

- Availability of daylight can be extended to account for external obstruction, different types of over hangs and side fins, and variable shapes of windows.
- The calculation of illumination on a reference point can take into account irregular shapes of rooms, the existence of partitions and furniture in the room.
- The solar energy calculation can be developed to accurately account for the thermal mass, time lag, and the different material in the room.
- Calculating the saving in energy can be done on a daily basis throughout the year using hourly data for external temperature, available solar energy and daylight. Calculation can also consider the internal heat gain from people and equipments.
- The computer program can be developed to accept graphical data and irregular room and fenestration shapes. It can be extended to show the user in three dimensions the location of the penetrated solar spot in a room at any time.
- More studies can be done on the sensitivity analysis of the fenestration design

elements such as changing the type and size of the overhang, the side fin, and the fenestration shape.

REFERENCES:-

[1] Carter, C., Villiers, J., "Principles of Passive Solar Buildings Design (with microcomputer programs)," Pergman Press,1985.

[2] The Commission of the European Communities, "Passive Solar Architecture in Europe 2," 1985.

[3] Rosenfeld, A., Selkowitz, S., "Beam Daylighting: Direct Use of Solar Energy for Interior Lighting," A Joint Conference of the American Society of the International Solar Energy Society and the Solar Energy Society of Canada Inc., Aug. 15th- 20th, Winnipeg, Vol. 7, 1976, pp. 375-391.

[4] Elvegard, E., Sjostedt, G., "The Calculation of Illumination from Sun and Sky," Illuminating Engineer, Vol. 35, 1940, pp. 333-342.

[5] Chroscicki, W., "Calculation Methods of Determining the Value of Daylight Intensity on the Ground of Photometrical and Actinometrical Measurements with Unobstructed Planes," Proceedings of the CIA Meeting, Session 71.24 (Barcelona, Spain). Rotterdam: Bouwcentrum International, 1971.

[6] Krockmann, J., Siedl, M., "Quantitative Data on Daylight for Illuminating Engineer," Lighting Research and Technology, Vol. 6, No. 3, pp. 65-171.

[7] IES Calculation Procedures Committee, "Recommended Practice for the Calculation of Daylight Availability," Journal of Illuminating Engineering Society, July 1984, pp. 381-392.

[8] Kittler, R., "Standardization of Outdoor Design Conditions for the Calculation of Daylight Factor with Clear Skies," Proceeding of The CIE conference on Sunlight in Buildings, 1967, pp. 273-285.

[9] Moon, P., Spencer, D., "Illumination from a Non-Uniform Sky," Illuminating Engineering, Vol. 37, December 1942, pp. 707-726.

[10] Gillette, G., Pierpoint, W., Treado, S., "A General Illuminance Model for Daylighting Availability," Journal of Illuminating Engineering Society, July 1984, pp. 330-339.

[11] Tregenza, P.R., "Cloud Cover and Daylight Illuminance," Building and Environment, Vol. 22, No. 3, 1987, pp. 163-165.

[12] Tregenza, P.R., "A Simple Mathematical Model of Illumination from a Cloudy Sky," Lighting Research and Technology, Vol. 12, No. 3, 1980, pp. 121-128.

[13] Tregenza, P.R., "Horizontal Illuminance from a Cloudy Sky," Building and Environment, Vol. 17, No. 3, 1982, pp. 217-222.

[14] Gillette, G., Kusuda, T., "A Daylighting Computation Procedure for Use in DOE-2 and Other Dynamic Building Energy Analysis Programs," Journal of Illuminating Engineering Society, January 1983, pp. 78-85.

[15] El Diasty, R., "A Study on Zenith Luminance in Montreal,Canada," Journal of Illuminating Engineering Society, Summer 1988, pp. 3-7.

[16] Karayel, M., Navvab, M., Ne'eman, E., and others, "Zenith Luminance and Sky Luminance Distributions for Daylighting Calculation," Energy and Buildings, Vol. 6, 1984, pp. 283-291.

[17] Zaki, N., "Building Fenestration, Daylighting," Ph.D. Thesis, Alexandria: Faculty of Engineering, University of Alexandria, 1986.

[18] Hopkinson, R., "Daylighting," Her Majesty's Stationary Office, London, 1963.

[19] Illuminating Engineering Society of North America, "IES Lighting Handbook: Reference Volume," 1981.

[20] Biesele, R., Arner, W., Conover, E., "A Lumen Method of Daylighting Design," Illuminating Engineer, Vol. 48, No. 1, pp. 39-45.

[21] Griffith, J., Arner, W., Conover, E., "A Modified Lumen Input Method of Daylighting Design," Illuminating Engineer, Vol. 50, No. 3, pp. 103-112.

[22] Dilaura, D.L., "On the Computation of Equivalent Sphere Illumination," Journal of Illuminating Engineering Society, January 1975, pp. 129-149.

[23] Dilaura, D.L., Hauser, G.A., "On Calculating the Effect of Daylighting in Interior Spaces," Journal of Illuminating Engineering Society, October 1978, pp. 2-14.

[24] El Diasty, R., Shapiro, M.M., Fazio, P., "A Time Dependent Concept for Daylighting Calculation; Part I: The Direct Component," Journal of Illuminating Engineering Society, Summer 1989, pp. 52-64.

[25] El Diasty, R., Shapiro, M.M, Fazio, P., "A Time Dependent Concept for Daylighting Calculation; Part II: The Reflected Component," Journal of Illuminating Engineering Society, Summer 1989, pp. 65-75.

[26] Nuretdinov, Kh.N., Nuritdinov, Sh.N., Kuchkarov, R.A., "Modelling Algorithm for the Process of Natural Illumination of Rooms with Consideration of the Opposite Buildings," Applied Solar Energy, Vol. 22, No. 4, 1986, pp. 79-84.

[27] Nuretdinov, Kh.N., Valiev, A., Nuritdinov, Sh.N., and others, "Study the Effect of an Opposite Buildings on Plane and Spatial Indices of Natural Illumination by Means of Computer Modelling," Applied Solar Energy, Vol. 23, No. 1, 1987, pp. 56-60.

[28] DiLaura, D.L., "On Radiation Transfer Calculation in Unempty Rooms," Journal of Illuminating Engineering Society, January 1981, pp. 120-126.

[29] DiLaura, D.L., "On a New Technique for Interreflected Component Calculations," Journal of Illuminating Engineering Society, October 1979, pp. 53-59.

[30] Lawrence Berkeley Laboratory, "Daylighting Design Tool Survey," SunWorld, Vol. 10, No. 4, 1986, pp. 110-114.

[31] Gillette, G., "A Daylighting Model for Building Energy Simulation," NBS Building Science Series 152, U.S. Department of Commerce, Washington, March 1983.

[32] Winkelmann, F.c., Selkowitz, S., "Daylighting Simulation in the DOE-2 Building Energy Analysis Program," Energy and Buildings, Vol. 8, 1985, pp. 271-286.

[33] Winkelmann, F.c., Selkowitz, S., "New Models for Analyzing the Thermal and Daylighting Performance of Fenestration," Proceedings of the ASHRAE/DOE Conference: Thermal Performance of The Exterior Envelopes of Buildings II, December 6-9 1982, Las Vegas, Nevada, pp. 597-616.

[34] Modest, M.F., "A General Model for the Calculation of Daylighting in Interior Spaces," Energy and Buildings, Vol. 5, 1982, pp. 69-79.

[35] Modest, M.F., "Daylighting Calculations for Non Rectangular Interior Spaces with Shading Devices," Journal of Illuminating Engineering Society, July 1983, pp. 226-241.

[36] Jordaans, A.A., "Computer Program for Calculating the Daylight Level in a Room," Energy and Buildings, Vol. 6, 1984, pp. 207-212.

[37] Bobenhausen, W., Lewis, S., "Daylighting Part I: Impacts of Daylighting on Electrical and Cooling Peak Demands in Commercial Buildings," Journal of Environmental Systems, Vol. 16, No.3, 1986-87, pp. 187-193.

[38] Building Research Station, "Lighting Control and Daylight Use," Digest No.272, April, 1983.

[39] Sanchez, N.E., Ruday, W., "Energy Impact of The Use of Daylighting in Offices," ASHRAE Transaction, Vol.87, Part 2, 1981, No. 2643.

[40] Oatman, P.A., Robbins, C.L., "Daylighting and Passive Solar Energy Systems in Commercial Buildings," Energy Engineering, Vol.86, No.1, 1989, pp.6-20.

[41] Leaver, J., McQueen, T., "Daylight Analysis with Microcomputers for School Buildings," Energy Engineering, Vol. 86, No.1, 1989, pp. 40-53.

[42] Gates, S., Wilcox, J., "Daylighting Analysis for Classrooms Using DOE-2.1B," Energy and Buildings, Vol. 6, 1984, pp. 331-341.

[43] Johnson, R., Sullivan, R., Selkowitz, S., and others, "Glazing Energy Performance and Design Optimization with Daylighting," Energy and Buildings, Vol. 6, 1984, pp. 305-317.

[44] Kittler, G.,"A Universal Calculation Method for Simple Predetermination of Natural Radiation on Building Surfaces and Solar Collectors," Building and Environment, Vol. 16, No.3, 1981, pp. 177-182.

[45] Illuminating Engineering Society of North America,"IES Lighting Handbook," 1981.

[46] Pierpoint, W., "A Simple Sky Model for Daylighting Calculations," General Proceedings, Edited by T. Vonier, International Daylight Conference, Phoenix, Az, Feb 16-18, 1983, pp.47.

[47] Nettleton, J., Murdoch, J., "Determination of Total External Illuminance on A Sloped Surface from Sunlight and Skylight," Journal of the Illuminating Engineering Society, July 1983, pp. 260-267.

[48] Robbins, C., "Daylight Design and Analysis," 1986, Van Nostrand Reinhold Company, New York.

[49] ASHRAE Handbook 1985 Fundamentals.

[50] Duffie, J., Beckman, W., "Solar Engineering of Thermal Process," 1980, John Wiley & Sons.

[51] Brandemuehl, M., Beckman, W., "Transmission of Diffuse Radiation Through CPC and Flat-Plate Collector Glazings," Solar Energy, Vol 24, 1980, pp. 511.

[52] Sun, T., "Shadow Area Equations for Window Overhangs and Side Fins and Their Application in Computer Calculation," ASHRAE Semiannual Meeting, Columbus, Ohio, Feb 5-8, 1968, No. 2059, pp. I.1.1-I.1.9.

[53] Rundquist, R., "Daylight Energy Saving Algorithm," ASHRAE Transaction, 1982, Part 2, No. 2685, pp. 343-373.

[54] McQuiston, F., Parker, J., "Heating, Ventilating, and Air Conditioning - Analysis and Design," 1982, John Wiley & Sons.

APPENDICES

APPENDIX A

CALCULATING THE SUN SPOT POSITION IN THE SPACE

In chapter III section 3.3.1 , the parameters and the flow chart required to determine where the sun spot will be on the space surfaces are shown. Detailed calculations for each case will be shown in this appendix to explain how the computation was achieved.

Fig (A.1) shows the variables used to determine the sun spot position. It is also important to refer to Fig (3.6).

Important notice:

The graphs shown in this appendix consider the sun rays flow from a direction as that shown in Fig (A.2a). If the sun direction changed to that shown in Fig (A.2b), the following corrections should be done:

1) $x1 = x$ instead of $x1 = length - m - x$.
2) *OX1* is replaced by *OX3*.
3) *OX3* is replaced by *OX1*.
4) *FX1* is replaced by *FX3*.
5) *FX3* is replaced by *FX1*.

CASE # 1

Definition: all the sun spot is on the floor.

Conditions: see flow chart Fig (3.22).

Position:

$$FX1 = length - x1 - m + z + (R / 2)$$

$$FX2 = m$$

$$FX3 = length - FX1 - FX2$$

$$FY3 = s$$

$$FY2 = s1 - s$$

$$FY1 = width - FY3 - FY2$$

Figure No. (3.7)

CASE # 2

Definition: all the sun spot is on the opposite wall.

Conditions: see flow chart Fig (3.22).

Parameters:

$$u2 = (P2 - width) \tan \gamma$$

$$u1 = (P1 - width) \tan \gamma$$

$$R' = u1 - u2$$

Position:

$$OX1 = length - (R'/2) - m - u2$$

$$OX2 = m$$

$$OX3 = u2 + (R'/2)$$

$$OZ1 = (s - width) \tan \varepsilon$$

$$OZ2 = (s1 - s) \tan \varepsilon$$

$$OZ3 = hight - OZ1 - OZ2$$

Figure No. (3.8)

CASE # 3

Definition: all the sun spot is on the perpendicular wall.

Conditions: see flow chart Fig (3.22).

Position:

$$SY3 = (P4 + P2) / 2 - (P3 - P2) / (2 \cos \varepsilon)$$

$$SY2 = (P3 - P2) / \cos \varepsilon)$$

$$SY1 = width - SY3 - SY2$$

$$SZ1 = (z1 + z2 - f) / 2$$

$$SZ2 = f$$

$$SZ3 = hight - sz1 - sz2$$

Figure No. (3.9)

CASE # 4

Definition: the sun spot is partly on the floor and the rest is on the opposite wall.

Conditions: see flow chart Fig (3.22).

Parameters:

$$MY1 = (P1 - width) \tan \gamma$$

$$R1 = R - ([s1 - width]\ R / [s1 - s])$$

$$R2 = MY1 + R1 + z - x1$$

Position:

$$FX1 = x + z + (R1/2)$$

$$FX2 = m$$

$$FX3 = x - z - (R1/2)$$

$$FY3 = s$$

$$FY2 = width - s$$

$$FY1 = 0$$

$$OX1 = length - (MY1 + m) + (R2/2)$$

$$OX2 = m$$

$$OX3 = MY1 - (R2/2)$$

$$OZ1 = 0$$

$$OZ2 = (s1 - width) \tan \varepsilon$$

$$OZ3 = hight - OZ2$$

Figure No. (3.10)

CASE # 5

Definition: point 3' is on the floor while points 1', 2' & 4' are on the perpendicular wall.

Conditions: see flow chart Fig (3.22).

Parameters:

$R1 = m + xl - z$

$AB = (sl - s) / (1 + [R - R1] / R1)$

$HHH = (P4 - s - AB) / \cos\psi$

$HA = AB \sin\psi$

$H = (3\ Z1 + Z2 + Z4) / 5$

$AREAS = (s - P2)(f) \cos(\lambda - \varepsilon) / \cos\varepsilon + (f + HHH)(HA) / 2$

$PAV = (P1 + P2 + P4) / 3$

$MZZ = AREAS / H$

Position:

$FX1 = length - (R1/2)$

$FX2 = R1 / 2$

$FX3 = 0$

$FY3 = s$

$FY2 = AB$

$FY1 = width - FY3 - FY2$

$SZ1 = Z2 / 2$

$SZ2 = H$

$SZ3 = hight - SZ1 - SZ3$

$SY3 = PAV - MZZ / 2$

$SY2 = MZZ$

$SY1 = width - SY3 - SY2$

Figure No. (3.11)

CASE # 6

Definition: points 3', 4' are on the floor while points 1', 2' are on the perpendicular wall.

Conditions: see flow chart Fig (3.22).

Position:

$FX1 = length - x1 - m + z + (R/2)$

$FX2 = x1 + m - z - (R/2)$

$FX3 = 0$

$FY3 = s$

$FY2 = s1 - s$

$FY1 = width - s1$

$SY3 = P2$

$SY2 = (s1 + s - P1 - P2)(s1 - s) \tan\varepsilon / (Z1 + Z2)$

$SY1 = width - SY3 - SY2$

$SZ1 = 0$

$SZ2 = (Z1 + Z2) / 2$

$SZ3 = hight - SZ2$

Figure No. (3.12)

CASE # 7

Definition: points 2', 3', 4' are on the floor while point 1' is on the perpendicular wall.

Conditions: see flow chart Fig (3.22).

Parameters:

$pp = s + (x1 - z)(s1 - s) \tan\varepsilon / (f \sin \psi \tan\gamma)$

$AREAS = m (pp - s) + [(m + (P4 - s1) \tan\gamma)(s1 - pp) / 2]$

$MZ = 3(s1 - pp) / 4$

Position:

$FX1 = length - x1 - m + z + (R/2)$

$FX2 = AREAS / (s1 - s)$

$FX3 = x1 - z - (R/2)$

$FY3 = s$

$FY2 = s1 - s$

$FY1 = width - s1$

$SY3 = (pp + s1 - MZ) / 2$

$SY2 = MZ$

$SY1 = width - SY3 - SY2$

$SZ1 = 0$

$SZ2 = 2/3\ Z1$

$SZ3 = hight - 2/3\ Z1$

Figure No. (3.13)

CASE # 8

Definition: the sun spot is on the floor, the perpendicular wall and the opposite wall (point 2' is on the floor and point 1' is on the perpendicular wall).

Conditions: see flow chart Fig (3.22).

Parameters:

$pp = s + (x1 - z)(s1 - s) \tan\varepsilon / (f \sin\psi \tan\gamma)$

$AA = R\ (width - pp) / (s1 - s)$

$H = (s1 - width) \tan\varepsilon$

$AREAS1 = 0.5\ (s1 - pp)(Z1) - 0.5\ (s1 - width)(H)$

$AREAS = m\ (pp - s) + (2m - AA)(width - pp) / 2$

$R4 = AREAS / (width - s)$

$R1 = R - ([s1 - width]\ R / [s1 - s])$

$AB = (P4 - width) \tan\gamma$

$AA1 = x1 + m - z - R1$

$H1 = 2\ Z1 / 3$

Position:

$FX1 = length - (x1 - z)/2 - R4$

$FX2 = R4$

$FX3 = (x1 - z) / 2$

$FY3 = s$

$FY2 = width - s$

$FY1 = 0$

$OX1 = length - (AB + AA1) / 2$

$OX2 = (AB + AA1) / 2$

$OX3 = 0$

$OZ1 = 0$

$OZ2 = H$

$OZ3 = hight - H$

$SZ1 = 0$

$SZ2 = H1$

$SZ3 = hight - H1$

$SY3 = width - AREAS1 / H1$

$SY2 = AREAS1 / H1$

$SY1 = 0$

Figure No. (3.14)

CASE # 9

Definition: the sun spot is on the floor, on the perpendicular wall and on the opposite wall (point 2' is on the floor and point 1' is on the opposite wall).

Conditions: see flow chart Fig (3.22).

Parameters:

$pp = s + (x1 - z)(s1 - s) \tan\varepsilon / (f \sin\psi \tan\gamma)$

$AA = R\,(width - pp) / (s1 - s)$

$H = (s1 - width) \tan\varepsilon$

$AREAS = m\,(pp - s) + (2m - AA)(width - pp) / 2$

$R1 = R - ([s1 - width]\, R / [s1 - s])$

$R2 = AB - m + AA$

$R4 = AREAS / (width - s)$

$AB = (P4 - width) \tan\gamma$

$AC = (AA)(H) / R1$

$H1 = 2\ AC / 3$

Position:

$FX1 = length - (x1 - z)/2 - R4$

$FX2 = R4$

$FX3 = (x1 - z) / 2$

$FY3 = s$

$FY2 = width - s$

$FY1 = 0$

$OX1 = length - AB + R2 / 2$

$OX2 = AB + R2 / 2$

$OX3 = 0$

$OZ1 = 0$

$OZ2 = H$

$OZ3 = hight - H$

$SY3 = width - [(AA)(H)(width - pp) / 2\ (R1)(H1)]$

$SY2 = [(AA)(H)(width - pp) / 2\ (R1)(H1)]OZ1 = 0$

$SY1 = 0$

$SZ1 = 0$

$SZ2 = H1$

$SZ3 = hight - H1$

Figure No. (3.15)

CASE # 10

Definition: the sun spot is on the floor, on the perpendicular wall and on the opposite wall (point 2' is not on the floor and point 1' is on the perpendicular wall).

Conditions: see flow chart Fig (3.22).

Parameters:

$H = (s1 - width) \tan\varepsilon$

$H1 = 2\ Z1\ /\ 3$

$R1 = R - ([s1 - width]\ R\ /\ [s1 - s])$

$R2 = m + x1 - z$

$R4 = R2 - R1\ /\ 2$

$AB = (P4 - width) \tan\gamma$

$AA1 = x1 + m - z - R1$

$AREAS = 0.5\ (Z1)(s1 - s) - 0.5\ (H)(s1 - width) + 0.5\ (f)(s - P2) \cos(\lambda - \varepsilon)\ /\ \cos\varepsilon$

Position:

$FX1 = length - R4$

$FX2 = R4$

$FX3 = 0$

$FY3 = s$

$FY2 = width - s$

$FY1 = 0$

$OX1 = length - (AB + AA1)\ /\ 2$

$OX2 = (AB + AA1)\ /\ 2$

$OX3 = 0$

$OZ1 = 0$

$OZ2 = H$

$OZ3 = hight - H$

$SZ1 = 0$

$SZ2 = H1$

$SZ3 = hight - H1$

$SY3 = width - AREAS\ /\ H1$

$SY2 = AREAS\ /\ H1$

$SY1 = 0$

Figure No. (3.16)

CASE # 11

Definition: the sun spot is on the floor, on the perpendicular wall and on the opposite wall (point 2' is not on the floor and point 1' is on the opposite wall).

Conditions: see flow chart Fig (3.22).

Parameters:

$H = (s1 - width) \tan\varepsilon$

$R1 = R - ([s1 - width] R / [s1 - s])$

$R2 = m + x1 - z$

$AB = (P1 - width) \tan\gamma$

$AA = m + x1 - z - R1$

$H1 = H (m - AA) / (m - AA + AB)$

$H2 = 2 (H1) / 3$

$AREAS = (m + AA - AB)(H) / 2 + 0.5 (m - AA + AB)(H) - 0.5 (m - AA)(H1)$

$R4 = (R2 + R1) / 2$

$R5 = AREAS / H$

$HH = (s - P2) \cos(\lambda - \varepsilon) / \cos\varepsilon$

$HHH = [(H1 - Z2)^2 + (width - P2)^2]^{0.5}$

$AREAS1 = 0.5 (width - s)(H1) + 0.5 (HHH)(HH)$

Position:

$FX1 = length - R4$

$FX2 = R4$

$FX1 = 0$

$FY3 = s$

$FY2 = width - s$

$FY1 = 0$

$OX1 = length - R5 - AB / 2$

$OX2 = R5$

$OX3 = AB / 2$

$OZ1 = 0$

$OZ2 = H$

$OZ3 = hight - H$

$SZ1 = 0$

$SZ2 = H2$

$SZ3 = hight - H2$

$SY3 = width - AREAS1 / H2$

$SY2 = AREAS1 / H2$

$SY1 = 0$

Figure No. (3.17)

CASE # 12

Definition: points 3', 4' are on the opposite wall while points 1', 2' are on the perpendicular wall.

Conditions: see flow chart Fig (3.22).

Parameters:

$H = (s1 - width) \tan\varepsilon$

$H1 = (s - width) \tan\varepsilon$

$AB = (P4 - width) \tan\gamma$

$AREAS = (H - H1)(width - P1) + 0.5\,(f)(P1 - P2) \cos(\lambda - \varepsilon) / \cos\varepsilon$

$H2 = (Z1 + Z2 + H1 + H) / 4$

$H3 = H - H1$

$AA = (P3 - width) \tan\gamma$

$R5 = (AB + AA) / 2$

Position:

$OX1 = length - R5$

$OX2 = R5$

$OX3 = 0$

$OZ1 = H1$

$OZ2 = H - H1$

$OZ3 = hight - H$

$SZ1 = H2 - H3 / 2$

$SZ2 = H3$

$SZ3 = HIGHT - SZ1 - SZ2$

$SY3 = width - (AREAS / H3)$

$SY2 = width - SY3$

$SY1 = width - SY2 - SY3$

Figure No. (3.18)

CASE # 13

Definition: points 1', 4' are on the opposite wall while points 2', 3' are on the perpendicular wall.

Conditions: see flow chart Fig (3.22).

Parameters:

$HH = (width - P2) / \cos\psi$

$HH1 = (width - P3) / \cos\psi$

$AREAS = [(HH + HH1) / 2] [(P3 - P2) / \cos\varepsilon] \cos(\lambda - \varepsilon)$

$MZZ = width - (P3 + P2) / 2$

$H = HH \sin\psi + Z2$

$H1 = HH1 \sin\psi + Z3$

$H2 = (H + H1 + Z3 + Z2) / 4$

$H3 = AREAS / MZZ$

$AB = (P4 - width) \tan\gamma$

$AA = (P1 - width) \tan\gamma$

$H6 = (s1 - width) \tan\varepsilon$

$AREAS1 = (AB - AA)(H6) + (H6 + H)(AA) / 2 - (H6 + H1)(AB) / 2$

$MZZ1 = H6 - (H + H1) / 2$

Position:

$OX1 = length - AREAS1 / MZZ1$

$OX2 = length - OX1$

$OX3 = 0$

$OZ1 = H6 - MZZ1$

$OZ2 = MZZ1$

$OZ3 = hight - H6$

$SZ1 = H2 - H3 / 2$

$SZ2 = H3$

$SZ3 = hight - SZ1 - SZ2$

$SY3 = width - MZZ$

$SY2 = MZZ$

$SY1 = 0$

Figure No. (3.19)

CASE # 14

Definition: point 4' are on the opposite wall while points 1', 2', 3' are on the perpendicular wall.

Conditions: see flow chart Fig (3.22).

Parameters:

$H = (s1 - width) \tan\varepsilon$

$HH = (z - H) / \sin\varepsilon$

$HH1 = (width - P3) / \cos\psi$

$H1 = HH1 \sin\psi + Z3$

$H2 = HH \cos(\varepsilon - \lambda)$

$H3 = (Z1 + Z2 + Z3 + H + H1) / 5$

$H4 = (Z1 - Z2 - Z3 + H) / 2$

$AREAS = (f)(H2) + (f + HH1)/2 \;\; [(\{P3 - P2\}/\cos\varepsilon) \cos(\lambda - \varepsilon) - H2]$

$MZZ = AREAS / H4$

$AB = (P4 - width) \tan\gamma$

$AREAS1 = 0.5 (H - H1)(AB)$

$R5 = AREAS1 / (H - H1)$

Position:

$OX1 = length - R5$

$OX2 = R5$

$OX3 = 0$

$OZ1 = H1$

$OZ2 = H - H1$

$OZ3 = hight - H$

$SZ1 = H3 - H4 / 2$

$SZ2 = H4$

$SZ3 = hight - SZ1 - SZ2$

$SY3 = width - MZZ$

$SY2 = MZZ$

$SY1 = 0$

Figure No. (3.20)

CASE # 15

Definition: points 1', 3', 4' are on the opposite wall while point 2' is on the perpendicular wall.

Conditions: see flow chart Fig (3.22).

Parameters:

$H = (s1 - width) \tan\varepsilon$

$H1 = (s - width) \tan\varepsilon$

$H3 = (width - P2) \tan(\pi/2 - \lambda) + (width - P2) \tan\varepsilon + H1$

$AB = (P3 - width) \tan\gamma$

$AREAS = m (H - H1) + (m + AB)(H3 - H1) / 2$

$AA = (P1 - width) \tan\gamma$

$R4 = AA / 2$

$R5 = AREAS / (H - H1)$

$AREAS1 = (H3 - H1)(width - P2) / 2$

$MZZ = 2 (width - P2) / 3$

$H4 = AREAS1 / MZZ$

$H5 = (H1 + Z2 + H3) / 3$

Position:

$OX1 = length - R4 - R5$

$OX2 = R5$

OX3 = R4

OZ1 = H1

OZ2 = H - H1

OZ3 = hight - H

SZ1 = H5 - H4 / 2

SZ2 = H4

SZ3 = hight - SZ1 - SZ2

SY3 = width - MZZ

SY2 = MZZ

SY1 = 0

Figure No. (3.21)

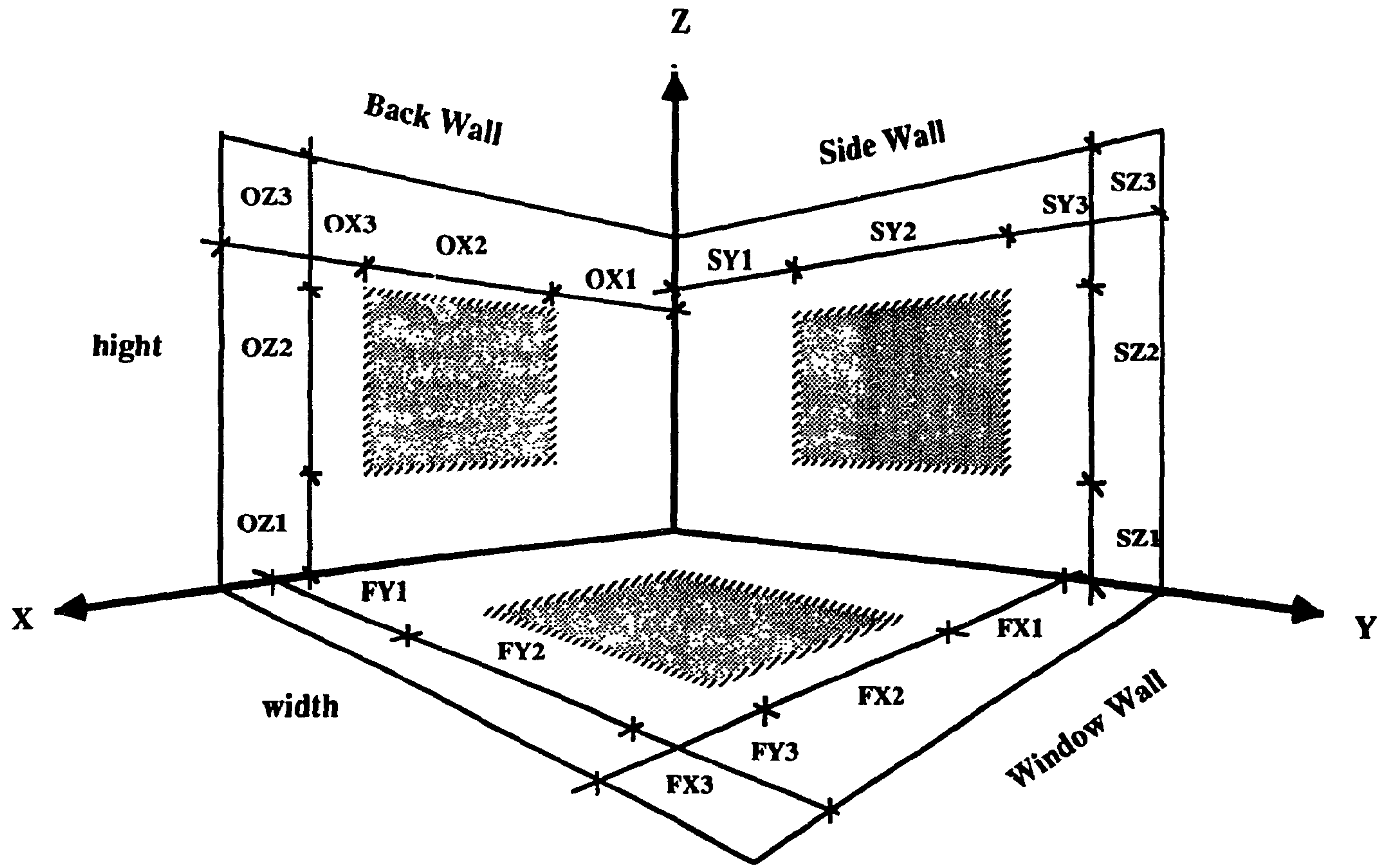


Fig A.1 Variables to determine the sun spot position in the space

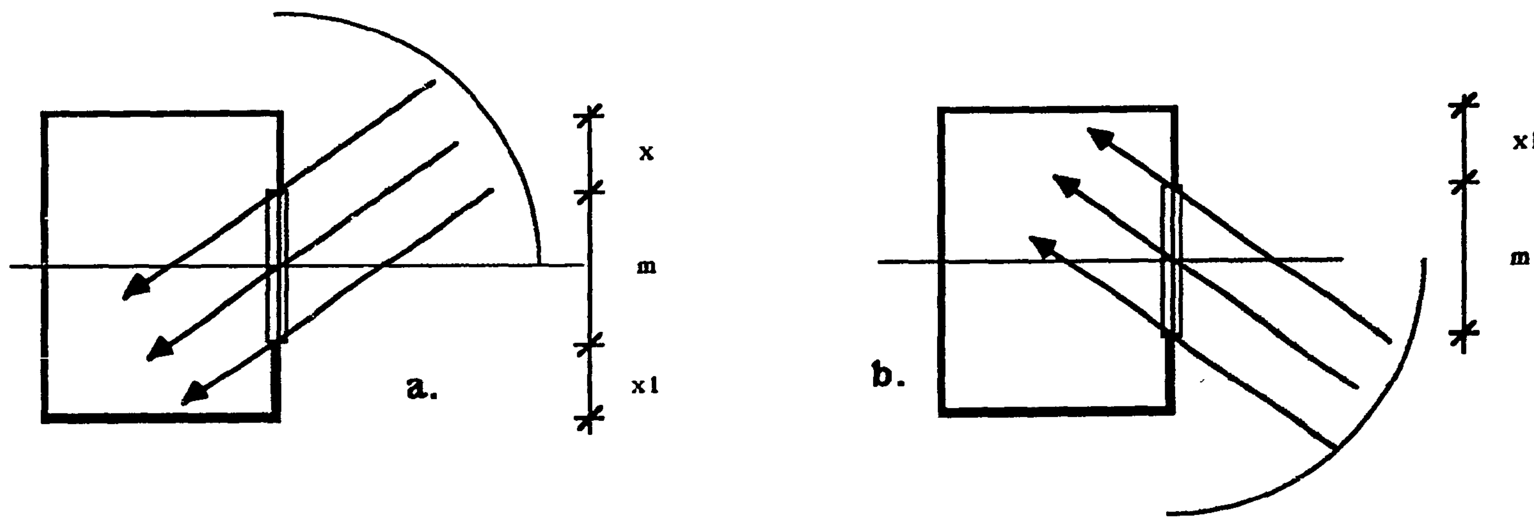


Fig A.2 Determination of the variable *x1*

APPENDIX B

COMPUTER PROGRAM LISTING

THE CONTROL MODULE

```
'*************************************************************************************************
'*************************************** D A Y L I G H T *****************************************
'*************************************************************************************************
'*********** PROGRAM DEVELOPED BY : A H M E D   M O K H T A R   IN 1990 AT CONCORDIA UNIVERSITY ***
'*************************************************************************************************

5       DIM VL.SS$(5), LO.SS%(5, 2), LE.SS%(5), TY.SS$(5), PIC.SS$(5), RG.SS(5, 2), CL.SS%(5, 2), SPECCHR.SS%(5): GOTO 15
10      DIM TENS(12), BTMSL(12), RBHLT(12), RBAUXIL(12), RBEREQ(1aw, BHLDT(12), BAUXIL(12), BEREQ(12), BHFLIGHT(12), BMLCONS(12), QS1(12):
        RETURN
12      DIM YTMSL(NSP), YTMHR(NSP), YTMRCL(NSP), YTENS(NSP): RETURN
15      KEY OFF: SD.SS% = 1: NUMSCR.SS% = 19: BLNK.SS$ = SPACE$(78)
        GOSUB 62890 'Test For Mono OR Color Display
        INIT.SS% = -1: SAME.SS% = 0
100     ON ERROR GOTO 62980    ' Error Handling Routine - Delete

'***********************************************************************************

3100    CLS : SCR.SS% = 1:   GOSUB 60000    'Introduction
3110    X.SS$ = INKEY$: IF X.SS$ <> CHR$(13) THEN 3110

3120    SCR.SS% = 2:   GOSUB 60000             'page 1
3125    X.SS$ = INKEY$:   IF RIGHT$(X.SS$, 1) = "I" THEN 3100
        IF X.SS$ <> CHR$(13) THEN 3125

3130    SCR.SS% = 3:   GOSUB 60000:   GOSUB 1100     'maim menu
        IF X.SS$ = "I" THEN 3120
        IF CHOICE = 0 THEN CLS : LOCATE 12, 25: PRINT "* THANK YOU FOR USING DAYLIGHT *": GOTO 3135
        ON CHOICE GOSUB 1500, 2000, 2500, 2600, 2650: GOTO 3130
3135    X.SS$ = INKEY$: IF X.SS$ <> CHR$(13) THEN 3135
        CLS : END

1100  '********** SUBROUTINE TO GET USER NEXT ACTION ********************
        INIT.SS% = -1:  SAME.SS% = 0
        IF LASTCHR.SS$ = "I" OR LASTCHR.SS$ = CHR$(13) THEN RETURN
1120    X.SS$ = INKEY$
        IF X.SS$ = "I" OR X.SS$ = CHR$(13) THEN RETURN ELSE 1120

'*******************************************************************.   ***************************************

1500    '************ NEW BUILDING DAT'  *********************
        ROUT.N% = 2: OPEN "ROUTF.AA.  ' FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        SHELL "B_DATA"
        RETURN

2000  '************ CHANGING BUILDING DATA *****************
        SCR.SS% = 17:   GOSUB 60000:   GOSUB 1100
        IF X.SS$ = "I" THEN 3130
        IF CHOICE = 0 THEN 3130
        ON CHOICE GOSUB 2100, 2150, 2200
        RETURN

2100  '************ CHANGE SPACE DATA *********************
        SCR.SS% = 14:  GOSUB 60000:   GOSUB 1100
        IF X.SS$ = "I" THEN GOTO 2000
        ROUT.N% = 3: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "INFO.A/A" FOR OUTPUT AS #1: PRINT #1, FIL$: PRINT #1, I: CLOSE
        SHELL "B_DATA"
        RETURN

2150  '************ CHANGE WINDOW DATA ********************
        SCR.SS% = 1:·  GOSUB 60000:   GOSUB 1100
        IF X.SS$ = "I" THEN GOTO 2000
        ROUT.N% = 4: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "INFO.AAA" FOR OUTPUT AS #1: PRINT #1, FIL$: PRINT #1, I, SW, SM, IDE.W$: CLOSE
        SHELL "B_DATA"
        RETURN
```

```
2200   ************* CHANGE SENSOR DATA *********************
       SCR.SS% = 16:  GOSUB 60000:  GOSUB 1100
       IF X.SS$ = "I" THEN GOTO 2000
       ROUT.N% = 5: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
       OPEN "INFO.AAA" FOR OUTPUT AS #1: PRINT #1, FIL$: PRINT #1, I, I3: CLOSE
       SHELL "B_DATA"
       RETURN

************************************************************************************************

2600   ************* NEW CITY DATA *************************
       ROUT.N% = 1: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
       SHELL "B_DATA"
       RETURN

2650   ************* CHANGING CITY DATA *********************
       LOCATE 22, 15: COLOR 14, 4: PRINT "       CITY NAME =              "
       LOCATE 22, 37: INPUT "", CITY.N$
       ROUT.N% = 6: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%, CITY.N$: CLOSE
       OPEN "INFO.AAA" FOR OUTPUT AS #1: PRINT #1, CITY.N$: CLOSE
       SHELL "B_DATA"
       RETURN

************************************************************************************************

2500   *********** SUBROUTINE TO RUN THE PROGRAM *************************
       SCR.SS% = 4:   GOSUB 60000:   GOSUB 1100
       IF X.SS$ = "I" THEN 3130
       IF CHOICE = 0 THEN 3130
       ON CHOICE GOSUB 4500, 3000
       RETURN
3000   *********** SUBROUTINE TO RUN FOR A SPECIFIC BUILDING **************
       SCR.SS% = 5:   GOSUB 60000:   GOSUB 1100
       IF X.SS$ = "I" THEN GOTO 2500
       IF CHOICE = 0 THEN 3130
       ON CHOICE GOSUB 3500, 4000
       RETURN

3500   *********** SUBROUTINE TO RUN FOR ONE MONTH ***********************
       LOCATE 21, 15: COLOR 14, 4: PRINT "      MONTH NO. =              "
       LOCATE 21, 37:  INPUT "", MON: MON = INT(MON)
       IF 1 > MON OR MON > 12 THEN 3500
       ROUT.N% = 1: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
       OPEN "DATA.AAA" FOR OUTPUT AS #1:    PRINT #1, MON, FIL$: CLOSE
       SHELL "D_LIGHT"
       OPEN "RESULT1.AAA" FOR INPUT AS #10:    ERASE TENS, BTMSL, RBHLT, RBAUXIL, RBEREQ, BHLDT, BAUXIL, BEREQ, BHFLIGHT,
       BMLCONS, QS1: GOSUB 10
       INPUT#10, MON, TENS(MON), BTMSL(MON), RBHLT(MON), RBAUXIL(MON), RBEREQ(MON), HFLIGHT, MLCONS, BHLDT(MON), BAUXIL(MON),
       BEREQ(MON), BHFLIGHT(MON), BMLCONS(MON), QS1(MON): CLOSE

3550   SCR.SS% = 12:   GOSUB 60000
       IF LASTCHR.SS$ = CHR$(13) THEN 3565
3560   X.SS$ = INKEY$: IF X.SS$ <> CHR$(13) THEN 3560
3565   IF CHOICE = 0 THEN 3130
       ON CHOICE GOSUB 3600, 3700
       RETURN

3600   *********** SUBROUTINE TO SHOW THE RESULT ON LOTUS *****************
       '(REQUIRE THE DRIVE NUMBER FOR THE LOTUS + THE DEFAULT DRIVE)

       LOCATE 22, 10: COLOR 14, 4: PRINT "                                  "
       LOCATE 22, 10: COLOR 14, 4: PRINT "    INPUT THE LOTUS DRIVE = "; ""
       LOCATE 22, 40: LINE INPUT "", DRV$:   XX$ = "CD " + DRV$
       LOCATE 23, 10: COLOR 14, 4: PRINT "    INPUT YOUR CURRUNT DRIVE = "; ""
       LOCATE 23, 43: LINE INPUT "", CDRV$:   XX3$ = "CD " + CDRV$
       IF DRV$ = "" OR CDRV$ = "" THEN 3550
       XX1$ = "COPY " + CDRV$ + "\MONTH.AHM"
       XX2$ = "COPY " + CDRV$ + "\LOTUS1.DAT"
       OPEN "LOTUS1.DAT" FOR OUTPUT AS #10:    PRINT #10, CDRV$: CLOSE
       XX4$ = MID$(XX$, 3, 3)
       XX5$ = MID$(XX3$, 3, 3)
       SHELL XX$
       SHELL XX4$
       SHELL XX1$
       SHELL XX2$
       SHELL "RENAME MONTH.AHM AUTO123.WK1"
       SHELL "123"
       SHELL "DEL AUTO123.WK1"
       SHELL XX3$
       SHELL XX5$
       GOTO 3550
```

```
3700  *********** SUBROUTINE TO MAKE A REPORT *************************
        SCR.SS% = 18:   GOSUB 60000
3710  X.SS$ = INKEY$:   IF RIGHT$(X.SS$, 1) = "!" THEN 3550
        IF X.SS$ <> CHR$(13) THEN 3710
        SCR.SS% = 22:   GOSUB 60000
3712  X.SS$ = INKEY$:   IF RIGHT$(X.SS$, 1) = "!" THEN 3700
        IF X.SS$ <> CHR$(13) THEN 3712
        GOTO 3550

4000  *********** SUBROUTINE TO RUN FOR A YEAR *************************
        ROUT.N% = 2: : OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "DATA.AAA" FOR OUTPUT AS #1:    PRINT #1, FIL$: CLOSE
        SHELL "D_LIGHT"
4050  SCR.SS% = 13:   GOSUB 60000
        IF LASTCHR.SS$ = CHR$(13) THEN 4065
4060  X.SS$ = INKEY$: IF X.SS$ <> CHR$(13) THEN 4060
4065  IF CHOICE = 0 THEN 3130
        ON CHOICE GOSUB 4100, 4150
        RETURN

4100  *********** SUBROUTINE TO SHOW THE RESULT ON GRAPHICS ****************

4105  SCR.SS% = 20:   GOSUB 60000
        IF LASTCHR.SS$ = CHR$(13) THEN 4050
4103  X.SS$ = INKEY$: IF X.SS$ <> CHR$(13) THEN 4103
        IF CHOICE = 0 THEN 4050
        ON CHOICE GOSUB 4110, 4115, 4120, 4125, 4130
        GOTO 4105

4110  RES$ = "RESULT1.AAA": GOSUB 4140
        SHELL "PLOT1"
        RETURN
4115  RES$ = "RESULT2.AAA": GOSUB 4140
        SHELL "PLOT1"
        RETURN
4120  RES$ = "RESULT3.AAA": GOSUB 4140
        SHELL "PLOT1"
        RETURN
4125  RES$ = "RESULT4.AAA": GOSUB 4140
        SHELL "PLOT1"
        RETURN
4130  RES$ = "RESULT5.AAA": GOSUB 4140
        SHELL "PLOT1"
        RETURN

4140   OPEN "RESUD.AAA" FOR OUTPUT AS #10: PRINT #10, RES$: CLOSE : RETURN

4150  *********** SUBROUTINE TO MAKE A REPORT *************************
        OPEN "RESULT.AAA" FOR INPUT AS #15: INPUT #15, T.SAV, BTMSLT, RBHLTT, RBAUXILT, RBEREQT, HFLIGHTT, MLCONST, BHLDTT,
        BAUXILT, BEREQT, BHFLIGHTT, BMLCONST, OS1T: CLOSE
4152  SCR.SS% = 19:   GOSUB 60000
4155  X.SS$ = INKEY$:   IF RIGHT$(X.SS$, 1) = "!" THEN 4050
        IF X.SS$ <> CHR$(13) THEN 4155
        SCR.SS% = 21:   GOSUB 60000
4157  X.SS$ = INKEY$:   IF RIGHT$(X.SS$, 1) = "!" THEN 4152
        IF X.SS$ <> CHR$(13) THEN 4157
        GOTO 4050

4500  *********** SUBROUTINE TO RUN FOR GENERAL INFORMATION ***************
        SCR.SS% = 6:   GOSUB 60000.   GOSUB 1100
        IF X.SS$ = "!" THEN GOTO 2500
        IF CHOICE = 0 THEN 3130
        ON CHOICE GOSUB 5000, 5100, 5250, 5350, 5400, 5500, 5600
        RETURN

5000  *********** SUBROUTINE FOR GENERAL INFORMATION (1) *****************
        SCR.SS% = 7:   GOSUB 60000:   GOSUB 1100
        IF X.SS$ = "!" THEN GOTO 4500
        ROUT.N% = 3: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "DATA.AAA" FOR OUTPUT AS #1:    PRINT #1, JULN, STT, LAT, LONGT, SKCO: CLOSE
        SHELL "D_LIGHT"
        OPEN "RESULT.AAA" FOR INPUT AS #1:   INPUT #1, EKH:  CLOSE : EKH = INT(EKH * 10) / 10
        MAG1$ = " SKY ILLUMINANCE ON":   MAG2$ = "A HORIZONTAL SURFACE =": MAG3$ = "        " + STR$(EKH) + " KLux"
        GOTO 6000
        RETURN

5100  *********** SUBROUTINE FOR GENERAL INFORMATION (2) *****************
        SCR.SS% = 8:   GOSUB 60000:   GOSUB 1100
        IF X.SS$ = "!" THEN GOTO 4500
```

```
        ROUT.N% = 4: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "DATA.AAA" FOR OUTPUT AS #1:   PRINT #1, JULN, STT, LAT, LONGT, SKCO, AZM: CLOSE
        SHELL "D_LIGHT"
        OPEN "RESULT.AAA" FOR INPUT AS #1:  INPUT #1, LWIND:  CLOSE : LWIND = INT(LWIND * 10) / 10
        MAG1$ = " SKY ILLUMINANCE ON":   MAG2$ = " A VERTICAL SURFACE =": MAG3$ = "     " + STR$(LWIND) + " KLux"
        GOTO 6000
        RETURN

5250  ************ SUBROUTINE FOR GENERAL INFORMATION (3) ******************
        SCR.SS% = 9:   GOSUB 60000:   GOSUB 1100
        IF X.SS$ = "I" THEN GOTO 4500
        ROUT.N% = 5: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "DATA.AAA" FOR OUTPUT AS #1:   PRINT #1, JULN, STT, LAT, LONGT, SKCO, AZM, SLOP: CLOSE
        SHELL "D_LIGHT"
        OPEN "RESULT.AAA" FOR INPUT AS #1:  INPUT #1, LWIND:  CLOSE : LWIND = INT(LWIND * 10) / 10
        MAG1$ = "  ILLUMINANCE ON":   MAG2$ = "  A TILTED SURFACE =": MAG3$ = "     " + STR$(LWIND) + " KLux"
        GOTO 6000
        RETURN

5350  ************ SUBROUTINE FOR GENERAL INFORMATION (4) ******************
        ROUT.N% = 6: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "DATA.AAA" FOR OUTPUT AS #1:   PRINT #1, JULN, STT, LAT, LONGT: CLOSE
        SHELL "D_LIGHT"
        OPEN "RESULT.AAA" FOR INPUT AS #1:  INPUT #1, HANG, SSHANG: CLOSE : HANG = INT(HANG * 10) / 10: SSHANG = INT(SSHANG * 10)
        / 10
        MAG1$ = " HOUR        SUNSET":   MAG2$ = " ANGLE     HOUR ANGLE": MAG3$ = STR$(HANG) + " deg. " + STR$(SSHANG) + " deg. "
        GOTO 6000

        RETURN

5400  ************ SUBROUTINE FOR GENERAL INFORMATION (5) ******************
        SCR.SS% = 9:   GOSUB 60000:   GOSUB 1100
        IF X.SS$ = "I" THEN GOTO 4500
        ROUT.N% = 7: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "DATA.AAA" FOR OUTPUT AS #1:   PRINT #1, JULN, STT, LAT, LONGT, SKCO, AZM, SLOP: CLOSE
        SHELL "D_LIGHT"
        OPEN "RESULT.AAA" FOR INPUT AS #1:  INPUT #1, EDT:  CLOSE : EDT = INT(EDT * 10) / 10
        MAG1$ = " SOLAR ILLUMINANCE =":   MAG2$ = "": MAG3$ = "      " + STR$(EDT) + " KLux"
        GOTO 6000
        RETURN

5500  ************ SUBROUTINE FOR GENERAL INFORMATION (6) ******************
        SCR.SS% = 10:  GOSUB 60000:   GOSUB 1100
        IF X.SS$ = "I" THEN GOTO 4500
        ROUT.N% = 8: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "DATA.AAA" FOR OUTPUT AS #1:   PRINT #1, JULN, STT, LAT, LONGT, AZM, SLOP: CLOSE
        SHELL "D_LIGHT"
        OPEN "RESULT.AAA" FOR INPUT AS #1:  INPUT #1, INANG: CLOSE : INANG = (INT(INANG * 10) / 10)
        MAG1$ = "  INCIDENT ANGLE":   MAG2$ = "   ON A SURFACE =": MAG3$ = "     " + STR$(INANG) + " Degree"
        GOTO 6000
        RETURN

5600  ************ SUBROUTINE FOR GENERAL INFORMATION (7) ******************
        ROUT.N% = 9: OPEN "ROUTF.AAA" FOR OUTPUT AS #1: PRINT #1, ROUT.N%: CLOSE
        OPEN "DATA.AAA" FOR OUTPUT AS #1:   PRINT #1, JULN, STT, LAT, LONGT: CLOSE
        SHELL "D_LIGHT"
        OPEN "RESULT.AAA" FOR INPUT AS #1:  INPUT #1, SALT, SAZMS: CLOSE
        SALT = (INT(SALT * 10) / 10): SAZMS = (INT(SAZMS * 10) / 10)
        MAG1$ = " SUN        SUN ":   MAG2$ = "ALTITUDE    AZIMUTH": MAG3$ = STR$(SALT) + " Deg.  " + STR$(SAZMS) + " Deg."
        GOTO 6000
        RETURN

6000  ************ SUBROUTINE FOR SHOWING THE RESULTS ********************
        SCR.SS% = 11:  GOSUB 60000
        LOCATE 10, 29: COLOR 7, 1: PRINT MAG1$: LOCATE 11, 29: COLOR 7, 1: PRINT MAG2$
        LOCATE 13, 29: COLOR 7, 1: PRINT MAG3$:  LOCATE 13, 38
6005  X.SS$ = INKEY$:   IF X.SS$ = CHR$(13) THEN 4500 ELSE 6005


'****************************************************************************************************
'****************************************** SCREENS DATA ********************************************
'****************************************************************************************************

        Copyright.SS$ = "(C)Copyright 84,85 The Software Bottling Company Of New York"
' ****************************************************************

30000 '
30005 '   Variables Section For B:INTROD
30020 RETURN
```

```
30025 '
30030 '  Assign VLSS$ Array to the variables
30045 RETURN
30050 '
30055 '  Section To Initialize Variables To Initial Values
30070 RETURN
30075 '
30080 '  **** List DATA statements & Print DISPLAY Only Variables ****
30085 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
30090 '
30095 RETURN
30100 '
30105 '  Screen Display Initialization Statements
30110 '
30115 NUMFLDS.SS% = 0: FILNM.SS$ = "INTROD.SCR":
30120 EXITCHR.SS$ = "I" + CHR$(127) + ""
30125 RESTORE 30080
30130 RETURN
30135 '
30140 '  Variables Section For B:PAGE_2
30155 RETURN
30160 '
30165 '  Assign VLSS$ Array to the variables
30180 RETURN
30185 '
30190 '  Section To Initialize Variables To Initial Values
30205 RETURN
30210 '
30215 '  **** List DATA statements & Print DISPLAY Only Variables ****
30220 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
30225 '
30230 RETURN
30235 '
30240 '  Screen Display Initialization Statements
30245 '
30250 NUMFLDS.SS% = 0: FILNM.SS$ = "PAGE_2.SCR":
30255 EXITCHR.SS$ = "I" + CHR$(127) + ""
30260 RESTORE 30215
30265 RETURN
30270 '
30275 '  Variables Section For B:MAIN
30280 '
30285 VLSS$(1) = STR$(CHOICE):
30290 RETURN
30295 '
30300 '  Assign VLSS$ Array to the variables
30305 '
30310 CHOICE = VAL(VLSS$(1)):
30315 RETURN
30320 '
30325 '  Section To Initialize Variables To Initial Values
30330 '
30335 CHOICE = 3:
30340 RETURN
30345 '
30350 '  **** List DATA statements & Print DISPLAY Only Variables ****
30355 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
30360 '
30365 DATA 72,18,1,"N","#",0,5,11,4,0
30370 RETURN
30375 '
30380 '  Screen Display Initialization Statements
30385 '
30390 NUMFLDS.SS% = 1: FILNM.SS$ = "MAIN.SCR":
30395 EXITCHR.SS$ = "I" + CHR$(127) + ""
30400 RESTORE 30350
30405 RETURN
30410 '
30415 '  Variables Section For B:RUNN_1
30420 '
30425 VLSS$(1) = STR$(CHOICE):
30430 RETURN
30435 '
30440 '  Assign VLSS$ Array to the variables
30445 '
30450 CHOICE = VAL(VLSS$(1)):
30455 RETURN
30460 '
30465 '  Section To Initialize Variables To Initial Values
30470 '
```

```
30475 CHOICE = 2:
30480 RETURN
30485 '
30490 '   **** List DATA statements & Print DISPLAY Only Variables ****
30495 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
30500 '
30505 DATA 72,18,1,"N","#",0,2,11,4,0
30510 RETURN
30515 '
30520 '   Screen Display Initialization Statements
30525 '
30530 NUMFLDS.SS% = 1: FILNM.SS$ = "RUNN_1.SCR":
30535 EXITCHR.SS$ = "I" + CHR$(127) + ""
30540 RESTORE 30490
30545 RETURN
30550 '
30555 '   Variables Section For B:RUNN_2
30560 '
30565 VL.SS$(1) = FIL$: VL.SS$(2) = STR$(CHOICE):
30570 RETURN
30575 '
30580 '   Assign VL.SS$ Array to the variables
30585 '
30590 FIL$ = VL.SS$(1): CHOICE = VAL(VL.SS$(2)):
30595 RETURN
30600 '
30605 '   Section To Initialize Variables To Initial Values
30610 '
30615 FIL$ = "": CHOICE = 1:
30620 RETURN
30625 '
30630 '   **** List DATA statements & Print DISPLAY Only Variables ****
30635 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
30640 '
30645 DATA 39,10,10,"C","XXXXXXXXXX",,,3,4,0
30650 DATA 72,18,1,"N","#",0,2,11,4,0
30655 RETURN
30660 '
30665 '   Screen Display Initialization Statements
30670 '
30675 NUMFLDS.SS% = 2: FILNM.SS$ = "RUNN_2.SCR":
30680 EXITCHR.SS$ = "I" + CHR$(127) + ""
30685 RESTORE 30630
30690 RETURN
30695 '
30700 '   Variables Section For B:GENERAL
30705 '
30710 VL.SS$(1) = STR$(JULN): VL.SS$(2) = STR$(STT): VL.SS$(3) = STR$(LAT):
30715 VL.SS$(4) = STR$(S6V5.SS):
30720 RETURN
30725 '
30730 '   Assign VL.SS$ Array to the variables
30735 '
30740 JULN = VAL(VL.SS$(1)): STT = VAL(VL.SS$(2)): LAT = VAL(VL.SS$(3)):
30745 CHOICE = VAL(VL.SS$(4)):
30750 RETURN
30755 '
30760 '   Section To Initialize Variables To Initial Values
30765 '
30770 JULN = 170: STT = 15.5: LAT = 45.54: S6V5.SS = 0:
30775 RETURN
30780 '
30785 '   **** List DATA statements & Print DISPLAY Only Variables ****
30790 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
30795 '
30800 DATA 70,12,3,"N","###",1,365,3,4,0
30805 DATA 70,16,4,"N","##.#",0.0,24.0,3,4,1
30810 DATA 70,19,6,"N","###.##",-90.00,90.00,3,4,1
30820 DATA 72,22,1,"N","#",0,7,3,4,0
30825 RETURN
30830 '
30835 '   Screen Display Initialization Statements
30840 '
30845 NUMFLDS.SS% = 4: FILNM.SS$ = "GENERAL.SCR":
30850 EXITCHR.SS$ = "I" + CHR$(127) + ""
30855 RESTORE 30785
30860 RETURN
30865 '
30870 '   Variables Section For B:IN_G_H
30875 '
```

```
30880 VL.SS$(1) = STR$(SKCO):
30885 RETURN
30890 '
30895 '   Assign VL.SS$ Array to the variables
30900 '
30905 SKCO = VAL(VL.SS$(1)):
30910 RETURN
30915 '
30920 '   Section To Initialize Variables To Initial Values
30925 '
30930 SKCO = 1:
30935 RETURN
30940 '
30945 '   **** List DATA statements & Print DISPLAY Only Variables ****
30950 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
30955 '
30960 DATA 45,10,1,"N","#",1,3,14,4,0
30965 RETURN
30970 '
30975 '   Screen Display Initialization Statements
30980 '
30985 NUMFLDS.SS% = 1: FILNM.SS$ = "IN_G_H.SCR":
30990 EXITCHR.SS$ = "I" + CHR$(127) + ""
30995 RESTORE 30945
31000 RETURN
31005 '
31010 '   Variables Section For B:IN_G_V
31015 '
31020 VL.SS$(1) = STR$(AZM): VL.SS$(2) = STR$(SKCO):
31025 RETURN
31030 '
31035 '   Assign VL.SS$ Array to the variables
31040 '
31045 AZM = VAL(VL.SS$(1)): SKCO = VAL(VL.SS$(2)):
31050 RETURN
31055 '
31060 '   Section To Initialize Variables To Initial Values
31065 '
31070 AZM = 0: SKCO = 1:
31075 RETURN
31080 '
31085 '   **** List DATA statements & Print DISPLAY Only Variables ****
31090 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31095 '
31100 DATA 52,7,3,"N","###",0,360,14,4,0
31105 DATA 45,13,1,"N","#",1,3,14,4,0
31110 RETURN
31115 '
31120 '   Screen Display Initialization Statements
31125 '
31130 NUMFLDS.SS% = 2: FILNM.SS$ = "IN_G_V.SCR":
31135 EXITCHR.SS$ = "I" + CHR$(127) + ""
31140 RESTORE 31085
31145 RETURN
31150 '
31155 '   Variables Section For B:IN_G_T
31160 '
31165 VL.SS$(1) = STR$(AZM): VL.SS$(2) = STR$(SLOP): VL.SS$(3) = STR$(SKCO):
31170 RETURN
31175 '
31180 '   Assign VL.SS$ Array to the variables
31185 '
31190 AZM = VAL(VL.SS$(1)): SLOP = VAL(VL.SS$(2)): SKCO = VAL(VL.SS$(3)):
31195 RETURN
31200 '
31205 '   Section To Initialize Variables To Initial Values
31210 '
31215 AZM = 0: SLOP = 0: SKCO = 1:
31220 RETURN
31225 '
31230 '   **** List DATA statements & Print DISPLAY Only Variables ****
31235 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31240 '
31245 DATA 51,8,3,"N","###",0,360,14,4,0
31250 DATA 52,9,2,"N","##",0,90,14,4,0
31255 DATA 45,13,1,"N","#",1,3,14,4,0
31260 RETURN
31265 '
31270 '   Screen Display Initialization Statements
31275 '
```

```
31280 NUMFLDS.SS% = 3: FILNM.SS$ = "IN_G_T.SCR":
31285 EXITCHR.SS$ = "I" + CHR$(127) + ""
31290 RESTORE 31230
31295 RETURN
31300 '
31305 '   Variables Section For B:IN_G_I
31310 '
31315 VL.SS$(1) = STR$(AZM): VL.SS$(2) = STR$(SLOP):
31320 RETURN
31325 '
31330 '   Assign VL.SS$ Array to the variables
31335 '
31340 AZM = VAL(VL.SS$(1)): SLOP = VAL(VL.SS$(2)):
31345 RETURN
31350 '
31355 '   Section To Initialize Variables To Initial Values
31360 '
31365 AZM = 0: SLOP = 0:
31370 RETURN
31375 '
31380 '   **** List DATA statements & Print DISPLAY Only Variables ****
31385 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31390 '
31395 DATA 52,9,3,"N","###",0,360,14,4,0
31400 DATA 52,15,2,"N","##",0,90,14,4,0
31405 RETURN
31410 '
31415 '   Screen Display Initialization Statements
31420 '
31425 NUMFLDS.SS% = 2: FILNM.SS$ = "IN_G_I.SCR":
31430 EXITCHR.SS$ = "I" + CHR$(127) + ""
31435 RESTORE 31380
31440 RETURN
31445 '
31450 '   Variables Section For B:INPUT_R
31465 RETURN
31470 '
31475          gn VL.SS$ Array to the variables
31490 RETURN
31495 '
31500 '   Section To Initialize Variables To Initial Values
31515 RETURN
31520 '
31525 '   **** List DATA statements & Print DISPLAY Only Variables ****
31530 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31550 RETURN
31555 '
31560 '   Screen Display Initialization Statements
31565 '
31570 NUMFLDS.SS% = 0: FILNM.SS$ = "INPUT_R.SCR":
31575 EXITCHR.SS$ = "I" + CHR$(127) + ""
31580 RESTORE 31525
31585 RETURN
31590 '   Variables Section For A:RESULT_M
31595 '
31600 VL.SS$(1)  STR$(I): VL.SS$(2) = STR$(CHOICE):
31605 RETURN
31610 '
31615 '   Assign VL.SS$ Array to the variables
31620 '
31625 I = VAL(VL.SS$(1)): CHOICE = VAL(VL.SS$(2)):
31630 RETURN
31635 '
31640 '   Section To Initialize Variables To Initial Values
31645 '
31650 I = 1: CHOICE = 2:
31655 RETURN
31660 '
31665 '   **** List DATA statements & Print DISPLAY Only Variables ****
31670 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31675 '
31680 DATA 71,14,2,"N","##",1,99,11,4,0
31685 DATA 72,20,1,"N","#",0,2,11,4,0
31690 RETURN
31695 '
31700 '   Screen Display Initialization Statements
31705 '
31710 NUMFLDS.SS% = 2: FILNM.SS$ = "RESULT_M.SCR":
31715 EXITCHR.SS$ = "I" + CHR$(127) + ""
31720 RESTORE 31665
```

```
31725 RETURN
31730 '
31735 '    Variables Section For A:RESULT_Y
31740 '
31745 VLSS$(1) = STR$(CHOICE):
31750 RETURN
31755 '
31760 '    Assign VLSS$ Array to the variables
31765 '
31770 CHOICE = VAL(VLSS$(1)):
31775 RETURN
31780 '
31785 '    Section To Initialize Variables To Initial Values
31790 '
31795 CHOICE = 2:
31800 RETURN
31805 '
31810 '    **** List DATA statements & Print DISPLAY Only Variables ****
31815 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31820 '
31830 DATA 72,20,1,"N","#",0,2,11,4,0
31835 RETURN
31840 '
31845 '    Screen Display Initialization Statements
31850 '
31855 NUMFLDS.SS% = 1: FILNM.SS$ = "RESULT_Y.SCR":
31860 EXITCHR.SS$ = "!" + CHR$(127) + ""
31865 RESTORE 31810
31870 RETURN
31875 '
31880 '    Variables Section For A:CHAN_S
31885 '
31890 VLSS$(1) = FIL$: VLSS$(2) = STR$(I):
31895 RETURN
31900 '
31905 '    Assign VLSS$ Array to the variables
31910 '
31915 FIL$ = VLSS$(1): I = VAL(VLSS$(2)):
31920 RETURN
31925 '
31930 '    Section To Initialize Variables To Initial Values
31935 '
31940 FIL$ = "": I = 1:
31945 RETURN
31950 '
31955 '    **** List DATA statements & Print DISPLAY Only Variables ****
31960 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31965 '
31970 DATA 46,10,6,"C","UUUUUU",,,0,4,0
31975 DATA 48,12,2,"N","##",1,99,0,4,0
31980 RETURN
31985 '
31990 '    Screen Display Initialization Statements
31995 '
32000 NUMFLDS.SS% = 2: FILNM.SS$ = "CHAN_S.SCR":
32005 EXITCHR.SS$ = "!" + CHR$(127) + ""
32010 RESTORE 31955
32015 RETURN
32020 '
32025 '    Variables Section For A:CHAN_W
32030 '
32035 VLSS$(1) = FIL$: VLSS$(2) = STR$(I): VLSS$(3) = STR$(SW):
32040 VLSS$(4) = STR$(SM): VLSS$(5) = IDE.W$:
32045 RETURN
32050 '
32055 '    Assign VLSS$ Array to the variables
32060 '
32065 FIL$ = VLSS$(1): I = VAL(VLSS$(2)): SW = VAL(VLSS$(3)): SM = VAL(VLSS$(4)):
32070 IDE.W$ = VLSS$(5):
32075 RETURN
32080 '
32085 '    Section To Initialize Variables To Initial Values
32090 '
32095 FIL$ = "": I = 1: SW = 1: SM = 1: IDE.W$ = "Y".
32100 RETURN
32105 '
32110 '    **** List DATA statements & Print DISPLAY Only Variables ****
32115 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32120 '
32125 DATA 46,7,6,"C","UUUUUU",,,0,4,0
```

```
32130 DATA 48,8,2,"N","##",1,99,0,4,0
32135 DATA 49,9,1,"N","#",1,4,0,4,0
32140 DATA 48,14,2,"N","##",1,10,0,4,0
32145 DATA 53,16,1,"Y","U",,,0,4,0
32150 RETURN
32155 '
32160 '   Screen Display Initialization Statements
32165 '
32170 NUMFLDS.SS% = 5: FILNM.SS$ = "CHAN_W.SCR":
32175 EXITCHR.SS$ = "I" + CHR$(127) + ""
32180 RESTORE 32110
32185 RETURN
32190 '
32195 '   Variables Section For A:CHAN_R
32200 '
32205 VL.SS$(1) = FIL$: VL.SS$(2) = STR$(I): VL.SS$(3) = STR$(I3)
32210 RETURN
32215 '
32220 '   Assign VL.SS$ Array to the variables
32225 '
32230 FIL$ = VL.SS$(1): : I = VAL(VL.SS$(2)): I3 = VAL(VL.SS$(3))
32235 RETURN
32240 '
32245 '   Section To Initialize Variables To Initial Values
32250 '
32255 FIL$ = "": I = 1: I3 = 1:
32260 RETURN
32265 '
32270 '   **** List DATA statements & Print DISPLAY Only Variables ****
32275 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32280 '
32285 DATA 46,10,6,"C","UUUUUU",,,0,4,0
32290 DATA 48,12,2,"N","##",1,99,0,4,0
32295 DATA 48,14,2,"N","##",1,10,0,4,0
32300 RETURN
32305 '
32310 '   Screen Display Initialization Statements
32315 '
32320 NUMFLDS.SS% = 3: FILNM.SS$ = "CHAN_R.SCR":
32325 EXITCHR.SS$ = "I" + CHR$(127) + ""
32330 RESTORE 32270
32335 RETURN
32340 '
32345 '   Variables Section For A:CHAN_C
32350 '
32355 VL.SS$(1) = STR$(CHOICE):
32360 RETURN
32365 '
32370 '   Assign VL.SS$ Array to the variables
32375 '
32380 CHOICE = VAL(VL.SS$(1)):
32385 RETURN
32390 '
32395 '   Section To Initialize Variables To Inital Values
32400 '
32405 CHOICE = 0:
32410 RETURN
32415 '
32420 '   **** List DATA statements & Print DISPLAY Only Variables ****
32425 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32430 '
32435 DATA 72,18,1,"N","#",0,3,11,4,0
32440 RETURN
32445 '
32450 '   Screen Display Initialization Statements
32455 '
32460 NUMFLDS.SS% = 1: FILNM.SS$ = "CHAN_C.SCR":
32465 EXITCHR.SS$ = "I" + CHR$(127) + ""
32470 RESTORE 32420
32475 RETURN
32480 '
32485 '   Variables Section For A:MRES
32490 '
32495 RETURN
32500 '
32505 '   Assign VL.SS$ Array to the variables
32510 '
32515 RETURN
32520 '
32525 '   Section To Initialize Variables To Inital Values
```

```
32530 '
32535 RETURN
32540 '
32545 '    **** List DATA statements & Print DISPLAY Only Variables ****
32550 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32555 '
32560 COLOR 1, 4: LOCATE 6, 74: PRINT USING "##"; MON;
32570 COLOR 14, 1: LOCATE 8, 26: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 10 - LEN(FIL$)), 6);
32575 COLOR 15, 1: LOCATE 11, 55: PRINT USING "####.#"; MLCONS / 1000;
32580 COLOR 15, 1: LOCATE 13, 55: PRINT USING "####.#"; BMLCONS(MON) / 1000;
32585 COLOR 15, 1: LOCATE 15, 55: PRINT USING "####.#"; RBHLT(MON) / 1000;
32590 COLOR 15, 1: LOCATE 17, 55: PRINT USING "####.#"; BHLDT(MON) / 1000;
32595 COLOR 15, 1: LOCATE 19, 55: PRINT USING "####.#"; OS1(MON) / 1000;
32600 RETURN
32605 '
32610 '    Screen Display Initialization Statements
32615 '
32620 NUMFLDS.SS% = 0: FILNM.SS$ = "MRES.SCR":
32625 EXITCHR.SS$ = "!" + CHR$(127) + ""
32630 RESTORE 32545
32635 RETURN
32640 '
32645 '    Variables Section For A:YRES
32650 '
32655 RETURN
32660 '
32665 '    Assign VL.SS$ Array to the variables
32670 '
32675 RETURN
32680 '
32685 '    Section To Initialize Variables To Initial Values
32690 '
32695 RETURN
32700 '
32705 '    **** List DATA statements & Print DISPLAY Only Variables ****
32710 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32715 '
32720 COLOR 14, 1: LOCATE 8, 26: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 10 - LEN(FIL$)), 6);
32730 COLOR 15, 1: LOCATE 11, 55: PRINT USING "####.#"; MLCONST;
32735 COLOR 15, 1: LOCATE 13, 55: PRINT USING "####.#"; BMLCONST;
32740 COLOR 15, 1: LOCATE 15, 55: PRINT USING "####.#"; RBHLTT;
32745 COLOR 15, 1: LOCATE 17, 55: PRINT USING "####.#"; BHLDTT;
32750 COLOR 15, 1: LOCATE 19, 55: PRINT USING "####.#"; OS1T;
32755 RETURN
32760 '
32765 '    Screen Display Initialization Statements
32770 '
32775 NUMFLDS.SS% = 0: FILNM.SS$ = "YRES.SCR":
32780 EXITCHR.SS$ = "!" + CHR$(127) + ""
32785 RESTORE 32705
32790 RETURN
32800 '
32805 '    Variables Section For D:RESULT1Y
32810 '
32815 VL.SS$(1) = STR$(CHOICE):
32820 RETURN
32825 '
32830 '    Assign VL.SS$ Array to the variables
32835 '
32840 CHOICE = VAL(VL.SS$(1)):
32845 RETURN
32850 '
32855 '    Section To Initialize Variables To Initial Values
32860 '
32865 CHOICE = 0:
32870 RETURN
32875 '
32880 '    **** List DATA statements & Print DISPLAY Only Variables ****
32885 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32890 '
32895 DATA 72,20,1,"N","#",0,5,11,4,0
32900 RETURN
32905 '
32910 '    Screen Display Initialization Statements
32915 '
32920 NUMFLDS.SS% = 1: FILNM.SS$ = "RESULT1Y.SCR":
32925 EXITCHR.SS$ = "!" + CHR$(127) + ""
32930 RESTORE 32880
32935 RETURN
32940 '
```

```
32945 '   Variables Section For D.MRES1
32950 '
32955 RETURN
32960 '
32965 '   Assign VL.SS$ Array to the variables
32970 '
32975 RETURN
32980 '
32985 '    Section To Initialize Variables To Initial Values
32990 '
32995 RETURN
33000 '
33005 '    **** List DATA statements & Print DISPLAY Only Variables ****
33010 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
33015 '
33020 COLOR 14, 1: LOCATE 8, 26: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 10 - LEN(FIL$)), 6);
33025 COLOR 15, 1: LOCATE 11, 55: PRINT USING "###.#"; RBAUXILT;
33030 COLOR 15, 1: LOCATE 13, 55: PRINT USING "###.#"; BAUXILT;
33035 COLOR 15, 1: LOCATE 15, 55: PRINT USING "###.#"; RBEREQT;
33040 COLOR 15, 1: LOCATE 17, 55: PRINT USING "###.#"; BEREQT;
33045 COLOR 15, 1: LOCATE 19, 55: PRINT USING "###.#"; T.SAV;
33050 RETURN
33055 '
33060 '   Screen Display Initialization Statements
33065 '
33070 NUMFLDS.SS% = 0: FILNM.SS$ = "YRES1.SCR":
33075 EXITCHR.SS$ = "!" + CHR$(127) + ""
33080 RESTORE 33005
33085 RETURN
33145 '   Variables Section For D:YRES1
33150 '
33155 RETURN
33160 '
33165 '   Assign VL.SS$ Array to the variables
33170 '
33175 RETURN
33180 '
33185 '    Section To Initialize Variables To Initial Values
33190 '
33195 RETURN
33200 '
33205 '    **** List DATA statements & Print DISPLAY Only Variables ****
33210 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
33215 '
33216 COLOR 1, 4: LOCATE 6, 74: PRINT USING "##"; MON;
33220 COLOR 14, 1: LOCATE 8, 26: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 10 - LEN(FIL$)), 6);
33225 COLOR 15, 1: LOCATE 11, 55: PRINT USING "###.#"; RBAUXIL(MON) / 1000;
33230 COLOR 15, 1: LOCATE 13, 55: PRINT USING "###.#"; BAUXIL(MON) / 1000;
33235 COLOR 15, 1: LOCATE 15, 55: PRINT USING "###.#"; RBEREQ(MON) / 1000;
33240 COLOR 15, 1: LOCATE 17, 55: PRINT USING "###.#"; BEREQ(MON) / 1000;
33245 COLOR 15, 1: LOCATE 19, 55: PRINT USING "###.#"; TENS(MON) / 1000;
33250 RETURN
33255 '
33260 '   Screen Display Initialization Statements
33265 '
33270 NUMFLDS.SS% = 0: FILNM.SS$ = "MRES1.SCR":
33275 EXITCHR.SS$ = "!" + CHR$(127) + ""
33280 RESTORE 33205
33285 RETURN
60000 '
60010 ' ┌──────────────────────────────────────────────────┐
60020 ' │ START OF SCREEN SCULPTOR FULL SCREEN EDITING ROUTINE │
60030 ' └──────────────────────────────────────────────────┘
60050 '              ┌─────────────────────┐
60060 '              │ Main body of subroutine │
60070 '              └─────────────────────┘
60080 '
60085  IF SAME.SS% THEN 60200  'To return to the SAME screen with SAME values
60090 'IF SCR.SS% = SCRLST.SS% THEN 60140' If same screen as last, don't reload
60100                          ' Get screen setup parameters
60105  ON SCR.SS% GOSUB 30105, 30240, 30380, 30520, 30665, 30835, 30975, 31120, 31270 31415, 31560, 31700, 31845, 31990, 32160, 32310, 32450,
       32610, 32765, 32910, 33060, 33260
60110    GOSUB 60550            ' Read field data for this screen
60120    OUT &H3D8, &H1          ' Turn off screen display
60125 ' CALL QUBLOAD(FILNM.SS$)    '<- For QuickBASIC, See READ.ME file
60130    DEF SEG = SCRNSEG.SS%: BLOAD FILNM.SS$, 0: DEF SEG ' Load screen picture
60135                          ' Set initial values
60140  IF INIT.SS% THEN ON SCR.SS% GOSUB 30055, 30190, 30325, 30465, 30605, 30760, 30920, 31060, 31205, 31355, 31500, 31640, 31785, 31930, 32085,
32245, 32395, 32525, 32685, 32855, 32985, 33185
60145                          ' Assign current values to screen array
```

```
60155   ON SCR.SS% GOSUB 30005, 30140, 30275, 30415, 30555, 30700, 30870, 31010, 31155, 31305, 31450, 31590, 31735, 31880, 32025, 3219' 32345,
32485, 32645, 32805, 32945, 33145
60160   OUT &H3D8, &H29              ' Turn on screen display
60170   GOSUB 60590                  ' Pad fields with blanks and display
60175                                ' Display initial DISPLAY variables
60180   ON SCR.SS% GOSUB 30080, 30215, 30350, 30490, 30630, 30785, 30945, 31085, 31230, 31380, 31525, 31665, 31810, 31955, 32110, 32270, 32420,
32545, 32705, 32880, 33005, 33205
60185   OUT &H3D8, &H29              ' Turn on screen display
60190 '
60195 F.SS% = 1: SCRLST.SS% = SCR.SS%
60200 COLOR 7, 0: LOCATE 25, 1: PRINT BLNK.SS$; 'Clr msg from prior screen
60205 IF NUMFLDS.SS% = 0 THEN RETURN          'Exit if no fields on screen
60210 LOCATE , , , 0, 1                     'Make cursor size large
60215 EXSCR.SS% = 0                          'Initialize Flag For Screen Exit
60220 WHILE NOT EXSCR.SS%         'Loop on each field(until Exit Flag is set)
60250     GOSUB 60740                      'Accept input data for this field
60260     '
60320 WEND
60360 '
60370 '
60400 FOR F.SS% = 1 TO NUMFLDS.SS%   ' Test Each Field Before Leaving Screen
60410     GOSUB 62500              ' Check contents of this field
60420     IF ERR.MSG% = -1 THEN EXSCR.SS% = 0: GOTO 60220'Error Detected
60430 NEXT F.SS%
60440 F.SS% = FLDLST.SS%                   'Reset Field indicator
60450 '
60455                                  'Assign new field values
60460 ON SCR.SS% GOSUB 30030, 30165, 30300, 30440, 30580, 30720, 30895, 31035, 31180, 31330, 31475, 31615, 31760, 31905, 32055, 32220, 32370,
    32505, 32665, 32830, 32965, 33165
60470 RETURN                           'Exit this subroutine
60480 '
60490 '*****************************************************************
60500 '
60510 '
60520 ' ***********************************
60530 ' **** Read Field Data For This Screen ****
60540 ' ***********************************
60550 FOR F.SS% = 1 TO NUMFLDS.SS%
60555     READ LO.SS%(F.SS%, 2), LO.SS%(F.SS%, 1), LE.SS%(F.SS%), TY.SS$(F.SS%), PIC.SS$(F.SS%), RG.SS(F.SS%, 1), RG.SS(F.SS%, 2),
        CL.SS%(F.SS%, 1), CL.SS%(F.SS%, 2), SPECCHR.SS%(F.SS%)
60560 NEXT F.SS%
60565 RETURN
60570 '
60575 '*****************************************************************
60580 '**** Pad Fields With Blanks, Insert Special Characters, and Display Flds
60585 '*****************************************************************
60590 FOR F.SS% = 1 TO NUMFLDS.SS%
60595    IF TY.SS$(F.SS%) = "N" THEN 60655 ' This section for non-numeric types
60600       IF LEN(VL.SS$(F.SS%)) > LE.SS%(F.SS%) THEN VL.SS$(F.SS%) = LEFT$(VL.SS$(F.SS%), LE.SS%(F.SS%)): GOTO 60610
60605       VL.SS$(F.SS%) = VL.SS$(F.SS%) + MID$(BLNK.SS$, 1, LE.SS%(F.SS%) - LEN(VL.SS$(F.SS%)))
60610       IF INSTR("CD", TY.SS$(F.SS%)) = 0 OR SPECCHR.SS%(F.SS%) = 0 THEN 60690
60615         CNT.SS% = 0
60620         FOR J.SS% = 1 TO LE.SS%(F.SS%)        ' Insert Special chars
60625              IF INSTR("ULX#89", MID$(PIC.SS$(F.SS%), J.SS%, 1)) = 0 THEN MID$(VL.SS$(F.SS%), J.SS%, 1) = MID$(PIC.SS$(F.SS%), J.SS%, 1):
        CNT.SS% = CNT.SS% + 1
60630            IF CNT.SS% = SPECCHR.SS%(F.SS%) THEN 60690
60635         NEXT J.SS%
60640       GOTO 60690        ' End of "non-numeric type" section
60645 '
60650 '                   ' The following section is for numeric types
60655    NUMDEC% = LE.SS%(F.SS%) - INSTR(PIC.SS$(F.SS%), ".")' Calc # of dec places
60657    IF LEFT$(VL.SS$(F.SS%), 1) = " " THEN VL.SS$(F.SS%) = RIGHT$(VL.SS$(F.SS%), LEN(VL.SS$(F.SS%)) - 1)' Strip leading blank
60660    IF NUMDEC% = LE.SS%(F.SS%) THEN NUMDEC% = 0: NUMINT% = LE.SS%(F.SS%) ELSE NUMINT% = LE.SS%(F.SS%) - NUMDEC% - 1
        ' Calc # of interger places
60665    IF VAL(VL.SS$(F.SS%)) = 0 THEN VL.SS$(F.SS%) = LEFT$(MID$(BLNK.SS$, 1, NUMINT% - 1) + "0." + STRING$(NUMDEC%, "0"), LE.SS%(F.SS%)):
        GOTO 60690' If no initial value
60670       DEC.VL% = INSTR(VL.SS$(F.SS%), ".")' IF DEC.VL% = 0 THEN DEC.VL% = LE.SS%(F.SS%) + 1          ' Position of decimal point in data
60675    VL.SS$(F.SS%) = LEFT$(RIGHT$(MID$(BLNK.SS$, 1, NUMINT%) + LEFT$(VL.SS$(F.SS%), DEC.VL% - 1), NUMINT%) + "." + MID$(VL.SS$(F.SS%),
        DEC.VL% + 1) + STRING$(NUMDEC%, "0"), LE.SS%(F.SS%))
60680 '
60685 '
60690 GOSUB 62310            ' Display Contents Of This Field
60695 NEXT F.SS%
60700 RETURN
60705 '
60710 '
60715 '      ******************************************
60720 '      *** Accept Input Data For The Current Field ***
60725 '      ******************************************
60730 '
```

```
60740 IF TY.SS$(F.SS%) <> "N" THEN A.SS% = 1: GOTO 60790
60750     NEWNUM% = -1: NUMED.SS% = 0 'Set Flags for num fld
60760     DECPOS% = INSTR(PIC.SS$(F.SS%), "."): IF DECPOS% = 0 THEN DECPOS% = LE.SS%(F.SS%) + 1
60770     A.SS% = DECPOS% - 1
60780 '                         ' Look for non-edit characters
60790 WHILE INSTR("ULX#98", MID$(PIC.SS$(F.SS%), A.SS%, 1)) = 0: A.SS% = A.SS% + 1: WEND
60800 CURCOL% = LO.SS%(F.SS%, 2) + A.SS% - 1' Find cursor position on screen
60810 LOCATE LO.SS%(F.SS%, 1), CURCOL%, 1' Starting position in this field
60820 '
60830 FLDLST.SS% = F.SS%            ' Reset field indicator
60840 EXFLD.SS% = 0                 ' Initialzei flag to exit this field
60850 WHILE NOT EXFLD.SS%           ' Loop while still editing this field
60860   X.SS$ = INKEY$: IF X.SS$ = "" THEN 60860' Wait for next keyboard character
60870   IF ERR.MSG% THEN ERR.MSG% = 0: COLOR 7, 0: LOCATE 25, 1, 0: PRINT STRING$(79, " ");          ' Erase old message line.
60880  IF LEN(X.SS$) > 1 OR INSTR(CHR$(8) + CHR$(13) + CHR$(27), X.SS$) <> 0 THEN X.SS$ = RIGHT$(X.SS$, 1) ELSE 60960 'Extented code key pressed?
60885     ON INSTR(";<=>?@ABCD", X.SS$) GOSUB 61100, 61100, 61100, 61100, 61100, 61100, 61100, 61100, 61100, 61100' This is a DUMMY statement.
          It traps the Function Keys (F1 - F10).  It is here to make user modifications simpler.
60887         ' For the above line to be active, you need to set all Function Key values to null. i.e.-  KEY 1,"" ; KEY 2,"" etc.
60890     IF TY.SS$(F.SS%) <> "N" THEN 60920    'If numeric, need special test
60900       IF INSTR("GO", X.SS$) <> 0 THEN 61030'Codes not valid for numeric
60910       IF INSTR("RKM" + CHR$(8), X.SS$) <> 0 THEN NUMED.SS% = -1: NEWNUM% = 0
60920     IF INSTR(EXITCHR.SS$, X.SS$) <> 0 THEN EXFLD.SS% = -1: EXSCR.SS% = -1: GOTO 61040'Check CODE to EXIT SCREEN
60930     ON INSTR("MKHPGRSO" + CHR$(8) + CHR$(13), X.SS$) GOSUB 61110, 61140, 61210, 61260, 61650, 61440, 61300, 61600, 61140, 61260: GOTO
61020
60940     GOTO 61030          ' Invalid extended code key pressed
60950 '
60960   IF TY.SS$(F.SS%) = "N" AND X.SS$ = "." THEN NEWNUM% = 0: NUMED.SS% = -1: GOTO 61010
60970   IF ASC(X.SS$) < 32 OR ASC(X.SS$) > 126 THEN 61030' Invalid characters
60980   GOSUB 61750: IF ERR.MSG% THEN 61030 ' Non-spec char entered: Test entry
60990   GOSUB 62230                 ' Add char to this field
61000   GOSUB 62310                 ' Print new field
61010   GOSUB 62380                 ' Move cursor to next position
61020   IF ERR.MSG% THEN 60740            ' If error re-edit this field
61030 WEND
61040 LASTCHR.SS$ = X.SS$                 ' Set last character indicator
61050 RETURN                      ' Exit from editing this field
61060 '
61070 '*************************************************************
61080 '****  THE FOLLOWING SUBROUTINES HANDLE EXTENDED KEYBOARD COMMANDS  ****
61090 '*************************************************************
61095 ' ***  DUMMY Subroutine for Function Keys (F1 - F10)
61096 ' ***  This is here for user modificatons   ***
61100 RETURN
61105 '*** Cursor right ***
61110    GOSUB 62380          ' Move cursor to next position
61120    RETURN
61130 '*** Cursor left or backspace ***
61140    IF CURCOL% = LO.SS%(F.SS%, 2) THEN 61210    'Goto prior field
61150    IF TY.SS$(F.SS%) = "N" AND INSTR(" +-", MID$(VL.SS$(F.SS%), A.SS%, 1)) <> 0 THEN RETURN
61160    CURCOL% = CURCOL% - 1: A.SS% = A.SS% - 1         'Pos of char in field
61170    IF INSTR("ULX#89", MID$(PIC.SS$(F.SS%), A.SS%, 1)) = 0 THEN 61140'Spec char?
61180    LOCATE LO.SS%(F.SS%, 1), CURCOL%, 1
61190    RETURN
61200 '*** Cursor up ***      Move to next field left
61210    GOSUB 62500                 ' Edit check field data before leaving
61220    IF ERR.MSG% THEN RETURN        ' Error found
61225    EXFLD.SS% = -1                 ' Set Flag to Exit this Field
61230    IF F.SS% > 1 THEN F.SS% = F.SS% - 1 ELSE F.SS% = NUMFLDS.SS%                    ' Goto prior fld
61240    RETURN
61250 '*** Cursor down or carriage return - Advance to next field ***
61260    GOSUB 62500                 ' Edit check field data before leaving
61270    IF ERR.MSG% THEN RETURN      ' Don't leave field if error was found
61275    EXFLD.SS% = -1               ' Set Flag to Exit this Field
61277    IF F.SS% = NUMFLDS.SS% AND INSTR(EXITCHR.SS$, CHR$(127)) <> 0 THEN EXSCR.SS% = -1'Test to leave screen after last field
61280    IF F.SS% < NUMFLDS.SS% THEN F.SS% = F.SS% + 1 ELSE F.SS% = 1                ' Increment fld num to next fld
61290    RETURN
61300 ' **** Del key pressed ******
61310 '            ' Start Del routine for Numeric fld on left of decimal pt
61320 IF TY.SS$(F.SS%) = "N" AND A.SS% < DECPOS% THEN MID$(VL.SS$(F.SS%), 1) = " " + LEFT$(VL.SS$(F.SS%), A.SS% - 1) + RIGHT$(VL.SS$(F.SS%),
          LE.SS%(F.SS%) - A.SS%): GOTO 61420
61330 '            ' Start Del routine for Numeric fld on right of decimal pt
61340 IF TY.SS$(F.SS%) = "N" THEN MID$(VL.SS$(F.SS%), 1) = LEFT$(VL.SS$(F.SS%), A.SS% - 1) + MID$(VL.SS$(F.SS%), A.SS% + 1, LE.SS%(F.SS%) -
          A.SS%) + "0": GOTO 61420
61350 '            ' Start Del routine for fld w/o non-edit chr
61360 IF SPECCHR.SS%(F.SS%) = 0 THEN MID$(VL.SS$(F.SS%), 1) = LEFT$(VL.SS$(F.SS%), A.SS% - 1) + MID$(VL.SS$(F.SS%), A.SS% + 1, LE.SS%(F.SS%)
          - A.SS%) + " ": GOTO 61420
61370 '
61380 CNT.SS% = 0              ' Start del routine for fld with non-edit chr
61390 WHILE INSTR("ULX#89", MID$(PIC.SS$(F.SS%), A.SS% + 1 + CNT.SS%, 1)) <> 0 AND CNT.SS% < LE.SS%(F.SS%) - A.SS%: CNT.SS% = CNT.SS%
          + 1: WEND' Count until next non-edit chr
```

```
61400 VL.SS$(F.SS%) = LEFT$(VL.SS$(F.SS%), A.SS% - 1) + MID$(VL.SS$(F.SS%), A.SS% + 1, CNT.SS%) + " " + RIGHT$(VL.SS$(F.SS%), LE.SS%(F.SS%)
        - A.SS% - CNT.SS%)' New value for field
61410 '
61420 CURCOL% = CURCOL% - 1: A.SS% = A.SS% - 1: GOSUB 62340: GOSUB 62380'Print fld; Set cursor
61430 RETURN
61440 ' ***** Ins key pressed ***
61450 '        ' Start Ins routine for numeric field on leftside of dec pt
61460 IF TY.SS$(F.SS%) = "N" AND A.SS% < DECPOS% THEN MID$(VL.SS$(F.SS%), 1) = MID$(VL.SS$(F.SS%), 2, A.SS% - 1) + "0" + RIGHT$(VL.SS$(F.SS%),
        LE.SS%(F.SS%) - A.SS%): GOTO 61580
61470 '        ' Start Ins routine for numeric field on rightside of dec pt
61480 IF TY.SS$(F.SS%) = "N" THEN VL.SS$(F.SS%) = LEFT$(VL.SS$(F.SS%), A.SS% - 1) + "0" + MID$(VL.SS$(F.SS%), A.SS%, LE.SS%(F.SS%) - A.SS%):
        GOTO 61580
61490 '        ' Start Ins routine for non-numeric w/o non-edit characters
61500 IF SPECCHR.SS%(F.SS%) = 0 THEN VL.SS$(F.SS%) = LEFT$(VL.SS$(F.SS%), A.SS% - 1) + " " + MID$(VL.SS$(F.SS%), A.SS%, LE.SS%(F.SS%) -
        A.SS%): GOTO 61580
61510 '        ' Start Ins routine for non-numeric with non-edit characters
61520 NEWVL$ = LEFT$(VL.SS$(F.SS%), A.SS% - 1) + " ": NEXTCHR$ = MID$(VL.SS$(F.SS%), A.SS%, 1)
61530 FOR I% = A.SS% + 1 TO LE.SS%(F.SS%)
61540   X.SS$ = MID$(PIC.SS$(F.SS%), I%, 1): IF INSTR("ULX#89", X.SS$) = 0 THEN NEWVL$ = NEWVL$ + X.SS$: GOTO 61570
61550   NEWVL$ = NEWVL$ + NEXTCHR$: NEXTCHR$ = MID$(VL.SS$(F.SS%), I%, 1)
61560 NEXT I%
61570 VL.SS$(F.SS%) = NEWVL$ + MID$(VL.SS$(F.SS%), I% + 1, LE.SS%(F.SS%))
61580 CURCOL% = CURCOL% - 1: A.SS% = A.SS% - 1: GOSUB 62310: GOSUB 62380' Print fld; Set cursor
61590 RETURN
61600 ' ***** END key pressed *****
61610 CURCOL% = LO.SS%(F.SS%, 2) + LE.SS%(F.SS%) - 1: A.SS% = LE.SS%(F.SS%)
61620 WHILE INSTR("ULX#89", MID$(PIC.SS$(F.SS%), A.SS%, 1)) = 0: A.SS% = A.SS% - 1: CURCOL% = CURCOL% - 1: WEND' Look for special protected
characters
61630 LOCATE LO.SS%(F.SS%, 1), CURCOL%, 1   ' Starting position in this field
61640 RETURN
61650 ' **** HOME key pressed ****  put cursor at beginning of field
61660 A.SS% = 1                        'Find cursor position for this field
61670 WHILE INSTR("ULX#89", MID$(PIC.SS$(F.SS%), A.SS%, 1)) = 0: A.SS% = A.SS% + 1: WEND' Look for special protected characters
61680 CURCOL% = LO.SS%(F.SS%, 2) + A.SS% - 1
61690 LOCATE LO.SS%(F.SS%, 1), CURCOL%, 1' Starting position in this field
61700 RETURN
61710 '
61715 '      ********************************************
61720 '      ****  ROUTINE TO EDIT-TEST CHARACTER ENTRY  ****
61730 '      ********************************************
61740 '***Check for special character type conversion***
61750 ON INSTR("NDMY", TY.SS$(F.SS%)) GOTO 61790, 62100, 62000, 61950
61760 ON INSTR("ULX#89", MID$(PIC.SS$(F.SS%), A.SS%, 1)) GOTO 61890, 61920, 62130, 61800, 62060, 62100
61770 PRINT "EDIT PICTURE TYPE "; MID$(PIC.SS$(F.SS%), A.SS%, 1); " NOT FOUND": STOP
61780 '
61790 '*** Numeric values; "."; "-"; "+"; " "
61800 ' NOTE: The decimal point is trapped in the"Accept input data" routine
61810 IF X.SS$ >= "0" AND X.SS$ <= "9" THEN RETURN
61820 IF X.SS$ <> "+" AND X.SS$ <> "-" AND X.SS$ <> " " THEN 61850' Is this a + or - sign?
61830 IF TY.SS$(F.SS%) = "C" THEN RETURN' +,-," " allowed anywhere in C type
61840   IF LEFT$(VL.SS$(F.SS%), A.SS% - 1) = SPACE$(A.SS% - 1) OR A.SS% = 1 THEN RETURN
61841 '     " " "+" and "-" sign only allowed in beginning of number
61850   MSG.SS$ = " Only numeric values can be entered here. Please re-enter. "
61860   GOSUB 62180: RETURN        ' Print error message; Exit
61870 '
61880 '*** Upper case or any other character***
61890 IF ASC(X.SS$) > 96 AND ASC(X.SS$) < 123 THEN X.SS$ = CHR$(ASC(X.SS$) - 32)
61900 GOTO 62130
61910 '*** Lower case or any other character***
61920 IF ASC(X.SS$) > 65 AND ASC(X.SS$) < 91 THEN X.SS$ = CHR$(ASC(X.SS$) + 32)
61930 GOTO 62130
61940 '*** Y/N answer only***
61950 IF X.SS$ = "Y" OR X.SS$ = "y" THEN X.SS$ = "Y": GOTO 62130
61960 IF X.SS$ = "N" OR X.SS$ = "n" THEN X.SS$ = "N": GOTO 62130
61970   MSG.SS$ = " Only 'Y' or 'N' can be entered here. Please re-enter. "
61980   GOTO 62180      ' Print error message
61990 '*** M/F answer only***
62000 IF X.SS$ = "M" OR X.SS$ = "m" THEN X.SS$ = "M": GOTO 62130
62010 IF X.SS$ = "F" OR X.SS$ = "f" THEN X.SS$ = "F": GOTO 62130
62020   MSG.SS$ = " Only 'M' or 'F' can be entered here. Please re-enter. "
62030   GOTO 62180      ' Print error message
62040 '
62050 ' *** Numeric values only ****
62060 IF (ASC(X.SS$) > 47 AND ASC(X.SS$) < 58) THEN GOTO 62130
62070   MSG.SS$ = " Only numeric values can be entered here. Please re-enter. "
62080   GOTO 62180      ' Print error message
62090 ' *** Numeric values and " " only ***
62100 IF (ASC(X.SS$) > 47 AND ASC(X.SS$) < 58) OR X.SS$ = " " THEN GOTO 62130
62110   MSG.SS$ = " Only numeric values or blanks can be entered here. Please re-enter. "
62120   GOTO 62180      ' Print error message
```

```
62130 RETURN
62140 '
62150 ' ******************************
62160 ' **** Print Error Messages *****
62170 ' ******************************
62180 ERR.MSG% = -1: SOUND 500, SD.SS% * 1: LOCATE 25, INT(81 - LEN(MSG.SS$)) / 2, 0: COLOR 0, 7: PRINT MSG.SS$; : LOCATE LO.SS%(F.SS%,
      1), CURCOL%, 1
62190 WHILE INKEY$ <> "": WEND   ' Clear input buffer after error
62200 RETURN
62210 '
62220 ' **************************************
62230 ' **** Add character to current field *****
62240 ' **************************************
62250 IF TY.SS$(F.SS%) = "N" AND NEWNUM% THEN MID$(VL.SS$(F.SS%), 1) = MID$(BLNK.SS$, 1, A.SS% - 1) + X.SS$ + "." + STRING$(LE.SS%(F.SS%),
      "0"): NEWNUM% = 0: RETURN' Restart fld - New num
62260 IF TY.SS$(F.SS%) <> "N" OR NUMED.SS% = 1 THEN MID$(VL.SS$(F.SS%), A.SS%, 1) = X.SS$: RETURN
62270 IF LEFT$(VL.SS$(F.SS%), 1) = " " THEN MID$(VL.SS$(F.SS%), 1, A.SS%) = MID$(VL.SS$(F.SS%), 2, A.SS% - 1) + X.SS$ ELSE NUMED.SS% = -1: GOTO
62260' Add character to left of decimal place
62280 RETURN
62290 '
62300 '******************************
62310 '**** Print new value of field ***
62320 '******************************
62330 '
62340 COLOR CL.SS%(F.SS%, 1), CL.SS%(F.SS%, 2)      ' Set color for this field
62350 LOCATE LO.SS%(F.SS%, 1), LO.SS%(F.SS%, 2), 0: PRINT VL.SS$(F.SS%); ' Print field
62360 RETURN
62365 '
62370 '**********************************
62380 '***** Move cursor to new location ****
62390 '**********************************
62400 IF TY.SS$(F.SS%) <> "N" OR NUMED.SS% <> 0 THEN 62420
62410   IF LEFT$(VL.SS$(F.SS%), 1) <> " " THEN NUMED.SS% = -1 ELSE A.SS% = DECPOS% - 1: CURCOL% = LO.SS%(F.SS%, 2) + A.SS% - 1: LOCATE
        LO.SS%(F.SS%, 1), CURCOL%, 1: RETURN
62415 '
62420 IF A.SS% < LE.SS%(F.SS%) THEN A.SS% = A.SS% + 1: CURCOL% = CURCOL% + 1 ELSE GOSUB 61260: RETURN' Advance cursor or go to next field
62430    IF INSTR("ULX#89", MID$(PIC.SS$(F.SS%), A.SS%, 1)) = 0 THEN 62420
62440    LOCATE LO.SS%(F.SS%, 1), CURCOL%, 1
62450 RETURN
62460 '
62470 ' **********************************
62480 ' *** Edit check final field result ***
62490 ' **********************************
62500 IF TY.SS$(F.SS%) <> "N" THEN 62540  ' Check numeric input range
62510   IF VAL(VL.SS$(F.SS%)) > RG.SS(F.SS%, 2) THEN MSG.SS$ = " The maximum value allowed in this field is " + STR$(RG.SS(F.SS%, 2)): GOSUB 62180:
        RETURN'Print Err Msg
62520   IF VAL(VL.SS$(F.SS%)) < RG.SS(F.SS%, 1) THEN MSG.SS$ = " The minimum value allowed in this field is " + STR$(RG.SS(F.SS%, 1)): GOSUB 62180:
        RETURN'Print Err Msg
62530 '
62540 IF TY.SS$(F.SS%) <> "D" THEN 62720      ' Date edit check
62550   IF VL.SS$(F.SS%) = "  /  /  " THEN 62720          ' No check
62560   DT.SS = VAL(RIGHT$(VL.SS$(F.SS%), 2) + LEFT$(VL.SS$(F.SS%), 2) + MID$(VL.SS$(F.SS%), 4, 2))
62570   IF INSTR("01,02,03,04,05,06,07,08,09,10,11,12", LEFT$(VL.SS$(F.SS%), 2)) <> 0 THEN 62590
62580     MSG.SS$ = "Invalid MONTH in date entered. Please re-enter.": GOSUB 62180: RETURN
62590   IF INSTR(MID$(VL.SS$(F.SS%), 4, 2), " ") = 0 AND VAL(MID$(VL.SS$(F.SS%), 4, 2)) > 0 AND VAL(MID$(VL.SS$(F.SS%), 4, 2)) < 32 THEN 62620
62600     MSG.SS$ = "Invalid DAY in date entered.  Please re-enter.": GOSUB 62180: RETURN
62610 '
62620   IF RG.SS(F.SS%, 1) = 0 AND RG.SS(F.SS%, 2) = 0 THEN 62720' No check
62630   IF RG.SS(F.SS%, 2) < RG.SS(F.SS%, 1) THEN 62680'Does date cross century?
62640     IF DT.SS >= RG.SS(F.SS%, 1) AND DT.SS <= RG.SS(F.SS%, 2) THEN 62720
62650       D1.SS$ = STR$(RG.SS(F.SS%, 1)): D2.SS$ = STR$(RG.SS(F.SS%, 2))
62660       MSG.SS$ = "The date should be between " + MID$(D1.SS$, 4, 2) + "/" + RIGHT$(D1.SS$, 2) + "/" + MID$(D1.SS$, 2, 2) + " and " + MID$(D2.SS$,
        4, 2) + "/" + RIGHT$(D2.SS$, 2) + "/" + MID$(D2.SS$, 2, 2): GOSUB 62180: RETURN
62670 '
62680   IF DT.SS >= RG.SS(F.SS%, 2) OR DT.SS <= RG.SS(F.SS%, 1) THEN 62720
62690       D1.SS$ = STR$(RG.SS(F.SS%, 2)): D2.SS$ = STR$(RG.SS(F.SS%, 1))
62700       MSG.SS$ = "The date should be between " + MID$(D1.SS$, 4, 2) + "/" + RIGHT$(D1.SS$, 2) + "/" + MID$(D1.SS$, 2, 2) + " and " + MID$(D2.SS$,
        4, 2) + "/" + RIGHT$(D2.SS$, 2) + "/" + MID$(D2.SS$, 2, 2): GOSUB 62180: RETURN
62710 '
62720 RETURN
62860 ' *********************************************
62870 ' ** Test If Monochrome or Color/Graphics Monitor **
62880 ' *********************************************
62890 DEF SEG = &H40
62900 MONO.SS = (PEEK(&H10) AND &H30) = &H30
62910 IF MONO.SS THEN SCRNSEG.SS% = &HB000 ELSE SCRNSEG.SS% = &HB800
62920 RETURN
62930 ' ***************************
62940 ' **  Error Handling Routine  **
62950 ' ***************************
```

```
62960 ' This routine is used mostly to turn the screen display back on if
62970 ' a sericus problem (such as the screen image file not found) occurs.
62980 CLS : OUT &H3D8, &H29     ' Turn screen display BACK ON (if off).
62990                    'The following checks for disk file errors
63000 IF NOT (ERR = 53 OR ERR = 57 OR ERR = 66 OR (ERR > 70 AND ERR < 77)) THEN 63030
63010 BEEP: COLOR 12: LOCATE 10, 10: PRINT "There is a problem finding a file on the disk.   Error code=" + STR$(ERR)
63020   LOCATE 13, 21: PRINT "(HINT:Check instalation of the program)": PRINT
          COLOR 9: LOCATE 24, 30: PRINT "HIT <RETURN> TO CONTINUE"
63025   X.CS$ = INKEY$:   IF X.SS$ <> CHR$(13) THEN 63025
63030   RESUME 63035
63035  END

'****************************************************************************************************
'****************************************************************************************************
'****************************************************************************************************
```

THE INPUT-DATA MODULE

```
10      DIM VL.SS$(53), LO.SS%(53, 2), LE.SS%(53), TY.SS$(53), PIC.SS$(53), RG.SS(53, 2), CL.SS%(53, 2), SPECCHR.SS%(53), CASE.S(4)
20      KEY OFF: SD.SS% = 1: NUMSCR.SS% = 12: BLNK.SS$ = SPACE$(78): INIT.SS% = -1: SAME.SS% = 0: LAST.SS% = 0
30      GOSUB 62890 'Test For Mono OR Color Display

        ROUTF$ = "ROUTF.AAA": OPEN ROUTF$ FOR INPUT AS #1: INPUT #1, ROUT.N%: CLOSE
150     DIM BIDA(7, 12), WEATH(13, 7): PI = 3.141592653#
        ON ROUT.N% GOSUB 500, 550, 1600, 1650, 1700, 1800: END
155     DIM RODA(NSP, 8, 8), WIDA(NSP, 10, 20), REFP(NSP, 10, 6): RETURN

********************************************************************************************
1100  ********** SUBROUTINE TO GET USER NEXT ACTION ********************************************
        INIT.SS% = -1:  SAME.SS% = 0: LAST.SS% = 0
        IF LASTCHR.SS$ = "I" OR LASTCHR.SS$ = CHR$(13) THEN RETURN
1120    X.SS$ = INKEY$
        IF X.SS$ = "I" OR X.SS$ = CHR$(13) THEN RETURN ELSE 1120

********************************************************************************************
************************** WEATHER DATA ****************************************************
********************************************************************************************

500   ****************** DATA FOR A NEW CITY ***************************************************
        SCR.SS% = 1: GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" THEN RETURN
        FOR I10 = 1 TO 12: WEATH(I10, 2) = 1 - WEATH(I10, 1) - WEATH(I10, 3): NEXT I10
        WEATH(13, 1) = WEATH(13, 1) * PI / 180: WEATH(13, 2) = WEATH(13, 2) * PI / 180: WEATH(13, 4) = WEATH(13, 4) * PI / 180
        OPEN CITY.N$ FOR OUTPUTawS #1
        FOR I10 = 1 TO 13: FOR I11 = 1 TO 7: PRINT #1, WEATH(I10, I11); : NEXT I11: PRINT #1, : NEXT I10: CLOSE
        RETURN

********************************************************************************************
************************** BUILDING DATA ***************************************************
********************************************************************************************

550   ****************** DATA FOR A NEW BUILDING ***********************************************
        SCR.SS% = 2: GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" THEN END
        IF INS.W$ = "Y" THEN BIDA(7, 9) = 12 ELSE BIDA(7, 9) = 24
        NSP = BIDA(1, 1): GOSUB 155:    FILL.N$ = FIL$

552     SCR.SS% = 13: GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" THEN ERASE RODA, WIDA, REFP: INIT.SS% = 0:  GOTO 550
        BIDA(1, 4) = 615: BIDA(1, 5) = .197: BIDA(1, 6) = -.147: BIDA(1, 7) = .681: BIDA(1, 8) = -.631: BIDA(1, 9) = 0:
        BIDA(1, 10) = .45: BIDA(1, 11) = -.4: BIDA(1, 12) = -.95:
        BIDA(5, 1) = 0: BIDA(5, 2) = .0026: BIDA(5, 3) = .0128: BIDA(5, 4) = .0085: BIDA(5, 5) = .0009: BIDA(5, 6) = 0:
        BIDA(5, 7) = 0: BIDA(5, 8) = .0248:
        BIDA(6, 1) = 0: BIDA(6, 2) = -1.7377: BIDA(6, 3) = .909: BIDA(6, 4) = -.11373: BIDA(6, 5) = .00496: BIDA(6, 6) = -.00001:

        FOR I = 1 TO NSP
553     GOSUB 555
554     NEXT I: GOSUB 1900:    GOSUB 1500: RETURN

555   ****************** DATA FOR A NEW SPACE DATA **********************************************
        RODA(I, 7, 2) = 0:    I2 = 0:    RODA(I, 7, 3) = 0.    RODA(I, 6, 5) = 0:
560     SCR.SS% = 3: GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" AND ROUT.N% = 3 THEN END
        IF X.SS$ = "I" AND I = 1 THEN INIT.SS% = 0:  GOTO 552
        IF X.SS$ = "I" THEN I = I - 1:    I3 = RODA(I, 7, 5): GOSUB 1455: GOTO 554
        IF INCL$ = "Y" THEN RODA(I, 7, 4) = 1 ELSE RODA(I, 7, 4) = 2
565     SCR.SS% = 4:  IF INCL$ = "Y" THEN WALL.N$ = "INCLINED WALL" ELSE WALL.N$ = "MAIN WALL"
        GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" THEN INIT.SS% = 0· GOTO 560
        ON RODA(I, 7, 4) GOTO 570, 575
570     SCR.SS% = 6· GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" THEN INIT.SS% = 0: GOTO 565
        GOTO 580
575     SCR.SS% = 5: GOSUB 60000. GOSUB 1100
        IF X.SS$ = "I" THEN INIT.SS% = 0awGOTO 565
580     SCR.SS% = 7: GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" AND RODA(I, 7, 4) = 1 THEN GOTO 570
        IF X.SS$ = "I" AND RODA(I, 7, 4) = 2 THEN GOTO 575
        GOSUB 400· GOSUB 5300:  I2 = 0:      I2.R% = 0
        FOR I10 = 1 TO 4: IF RODA(I, I10, 5) <> 0 THEN I2.R% = I2.R% + 1: CASE.S(I2.R%) = I10
        NEXT I10: I2 R% = 0
585     IF RODA(I, 7, 4) = 1 THEN WALL.N$ = "INCLINED WALL" ELSE WALL.N$ = "MAIN WALL"
        WD = RODA(I, 1, 5). I1 = 1: GOSUB 900
587     WALL.N$ = "RIGHT WALL": WD = RODA(I, 2, 5): I1 = 2: GOSUB 900
```

```
590     WALL.N$ = "OPPOSITE WALL": WD = RODA(I, 3, 5): I1 = 3: GOSUB 900
595     WALL.N$ = "LEFT WALL": WD = RODA(I, 4, 5): I1 = 4: GOSUB 900
600     GOSUB 1450:   'REFERENCE POINTS
        RETURN

400     ''''''''''''''''''''' COMPLETING THE DATA FOR THE SPACE '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        RODA(I, 1, 3) = RODA(I, 1, 3) * PI / 180:     RODA(I, 4, 1) = RODA(I, 2, 1)
        IF RODA(I, 1, 3) > PI THEN RODA(I, 1, 3) = RODA(I, 1, 3) - 2 * PI
        IF RODA(I, 7, 4) = 2 THEN RODA(I, 1, 6) = 90
        RODA(I, 1, 6) = RODA(I, 1, 6) * PI / 180
        RODA(I, 3, 1) = RODA(I, 1, 1): RODA(I, 5, 1) = RODA(I, 1, 1): RODA(I, 5, 2) = RODA(I, 2, 1)
        RODA(I, 1, 2) = RODA(I, 3, 2) / SIN(RODA(I, 1, 6)):        RODA(I, 2, 2) = RODA(I, 3, 2): RODA(I, 4, 2) = RODA(I, 3, 2)
        RODA(I, 3, 4) = RODA(I, 1, 4): RODA(I, 2, 4) = RODA(I, 1, 4): RODA(I, 4, 4) = RODA(I, 1, 4)
        RODA(I, 6, 1) = RODA(I, 1, 1): RODA(I, 6, 2) = RODA(I, 2, 1) - RODA(I, 3, 2) / TAN(RODA(I, 1, 6))
        RODA(I, 3, 3) = RODA(I, 1, 3) + PI:     RODA(I, 2, 3) = RODA(I, 1, 3) + PI / 2: RODA(I, 4, 3) = RODA(I, 1, 3) - PI / 2
        IF RODA(I, 3, 3) > PI THEN RODA(I, 3, 3) = RODA(I, 3, 3) - 2 * PI
        IF RODA(I, 2, 3) > PI THEN RODA(I, 2, 3) = RODA(I, 2, 3) - 2 * PI
        IF RODA(I, 4, 3) < -PI THEN RODA(I, 4, 3) = RODA(I, 4, 3) + 2 * PI
        RODA(I, 1, 7) = RODA(I, 1, 1) * RODA(I, 1, 2):        RODA(I, 3, 7) = RODA(I, 3, 1) * RODA(I, 3, 2)
        RODA(I, 2, 7) = (RODA(I, 2, 1) - (RODA(I, 3, 2) / TAN(RODA(I, 1, 6))) / 2) * RODA(I, 3, 2)
        RODA(I, 4, 7) = RODA(I, 2, 7):        RODA(I, 5, 7) = RODA(I, 5, 1) * RODA(I, 5, 2)
        RODA(I, 6, 7) = RODA(I, 6, 1) * RODA(I, 6, 2)
        RODA(I, 7, 2) = RODA(I, 1, 7) + RODA(I, 3, 7) + RODA(I, 2, 7) + RODA(I, 4, 7)
        RODA(I, 7, 3) = RODA(I, 1, 5) + RODA(I, 3, 5) + RODA(I, 2, 5) + RODA(I, 4, 5)
        RETURN

900     ''''''''''''''''' SUBROUTINE TO ENTER SEVERAL WINDOW DATA ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        IF WD = O THEN RETURN
        IF IDENT.W$ = "Y" THEN REP% = WD - 1: WD = 1
        I4 = 1: FOR I4 = 1 TO WD
905     I2 = I2 + 1:   GOSUB 910:   NEXT I4.  I2.R% = I2.R% + 1:   I1.R% = I1:   WALL.NR$ = WALL.N$
        IF IDENT.W$ = "Y" THEN GOTO 907 ELSE RETURN
907     FOR I10 = 1 TO REP%: FOR I11 = 1 TO 20: WIDA(I, I2 + I10, I11) = WIDA(I, I2, I11): NEXT I11: NEXT I10
        I2 = I2 + REP%: RETURN
908     I3.R% = CASE.S(I2.R%): I2.R% = I2.R% - 1: ON I3.R% GOTO 585, 587, 590:  ' TO DETERMINE WHERE TO GO IN CASE OF [PgUp]

910     ''''''''''''''''' SUBROUTINE TO ENTER ONE WINDOW DATA ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        WIDA(I, I2, 1) = I1
        IF RODA(I, 7, 4) = 1 THEN WIDA(I, I2, 6) = PI / 2 - RODA(I, 1, 6) ELSE WIDA(I, I2, 6) = 0
        WIDA(I, I2, 19) = RODA(I, I1, 3)
915     SCR.SS% = 8: GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" AND ROUT.N% = 4 THEN END
        IF X.SS$ = "I" AND I4 = 1 AND I2.R% = 0 THEN GOTO 580
        IF X.SS$ = "I" AND I4 = 1 THEN I2 = I2 - RODA(I, I1.R%, 5) - 1: GOTO 908
        IF X.SS$ = "I" THEN I4 = I4 - 1: I2 = I2 - 1: GOTO 905
        WIDA(I, I2, 18) = WIDA(I, I2, 18) / 100
        IF CHOICE1$ = "N" THEN 920
917     SCR.SS% = 9: GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" THEN INIT.SS% = 0: GOTO 915
920     IF CHOICE2$ = "N" THEN 925
        SCR.SS% = 10: GOSUB 60000. GOSUB 1100
        IF X.SS$ = "I" AND CHOICE1$ = "N" THEN INIT.SS% = 0: GOTO 915
        IF X.SS$ = "I" AND CHOICE1$ = "Y" THEN INIT.SS% = 0: GOTO 915
925     RETURN

1450    ''''''' SUBROUTINE TO INPUT SEVERAL REFERENCE POINT DATA ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

        FOR I3 = 1 TO RODA(I, 7, 5)
1455    GOSUB 1460:   NEXT I3
        RETURN

1460    ''''''' SUBROUTINE TO INPUT ONE REFERENCE POINT DATA ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        IF INCL$ = "Y" THEN WALL.N$ = "INCLINED WALL" ELSE WALL.N$ = "MAIN WALL"
        SCR.SS% = 11: GOSUB 60000: GOSUB 1100
        IF X.SS$ = "I" AND ROUT.N% = 5 THEN END
        IF X.SS$ = "I" AND I3 = 1 THEN I1 = I1.R%: I4 = I4 - 1: I2 = I2 - RODA(I, I1.R%, 5) + 1: WALL.N$ = WALL.NR$: I2.R% = I2.R% - 1: I3 = 0: GOTO
910     IF X.SS$ = "I" THEN INIT.SS% = 0: I3 = I3 - 1: GOTO 1455
        RETURN

1900    ''''''' SUBROUTINE TO INPUT THE END SCREEN '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
        SCR.SS% = 12: GOSUB 60000: IF I4.R% = 1 THEN I = I + 1
1910    X.SS$ = INKEY$: IF X.SS$ = "" THEN 1910
        IF RIGHT$(X.SS$, 1) = "I" THEN INIT.SS% = 0:    I = I - 1:    I3 = RODA(I, 7, 5): I4.R% = 1: GOSUB 1455: GOTO 1900
        IF X.SS$ = CHR$(13) THEN RETURN ELSE 1910

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''' CHANGING A SPACE DATA ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'    req: FIL$,I
```

```
1600      INFOF$ = "INFO.AAA":     OPEN INFOF$ FOR INPUT AS #1: INPUT #1, FIL$, I: CLOSE
          GOSUB 1550: INIT.SS% = -1:     GOSUB 555:     GOSUB 1500:     RETURN

'*******************************************************************************************
'************************* CHANGING A WINDOW DATA ******************************************
'*******************************************************************************************
'     req: FIL$, I, SW(WALL #), SM(WINDOW # IN THE WALL), IDE.W$(CHANGE ALL THE IDENTICAL WINDOWS)

1650      INFOF$ = "INFO.AAA":     OPEN INFOF$ FOR INPUT AS #1: INPUT #1, FIL$, I, SW, SM, IDE.W$: CLOSE
          GOSUB 1550: INIT.SS% = 0: WIDA(I, I2, 18) = WIDA(I, I2, 18) * 100
          IF IDE.W$ = "Y" THEN REP% = RODA(I, SW, 5) - 1: SM = 1
          IF SW = 1 THEN I2 = SM: GOTO 1655
          I2 = 0: FOR I5 = 1 TO SW - 1: I2 = I2 + RODA(I, SW, 5): NEXT I5: I2 = I2 + SM
1655      GOSUB 910
          IF IDE.W$ = "Y" THEN GOSUB 1660
          GOSUB 1500:     RETURN
1660      FOR I10 = 1 TO REP%: FOR I11 = 1 TO 20: WIDA(I, I2 + I10, I11) = WIDA(I, I2, I11): NEXT I11: NEXT I10: RETURN

'*******************************************************************************************
'************************* CHANGING A REFERENCE POINT DATA *********************************
'*******************************************************************************************
'     req: FIL$, I, I3

1700      INFOF$ = "INFO.AAA":     OPEN INFOF$ FOR INPUT AS #1: INPUT #1, FIL$, I, I3: CLOSE
          GOSUB 1550: INIT.SS% = 0:     GOSUB 1460:     GOSUB 1500:     RETURN

'*******************************************************************************************
'************************* CHANGING A CITY DATA ********************************************
'*******************************************************************************************
'     req: CITY.N$

1800      INFOF$ = "INFO.AAA":     OPEN INFOF$ FOR INPUT AS #1: INPUT #1, CITY.N$: CLOSE
          OPEN CITY.N$ FOR INPUT AS #1
          FOR I10 = 1 TO 13: FOR I11 = 1 TO 7: INPUT #1, WEATH(I10, I11): NEXT I11: NEXT I10: CLOSE
          INIT.SS% = 0:   GOSUB 500:   RETURN

'*******************************************************************************************
'*******************************************************************************************
'*******************************************************************************************

5300  '******************* SUBROUTINE TO CALCULATE EFFECTIVE ABSORPTANCE OF THE ROOM-WINDOW COMBINATION ************

          RRS = 0
          FOR I1 = 1 TO 6
          RRS = RRS + RODA(I, I1, 7) * RODA(I, I1, 4)
          NEXT I1
          RODA(I, 7, 1) = RRS / (RODA(I, 5, 7) + RODA(I, 7, 2) + RODA(I, 6, 7))
          RETURN

'***********************************************************************
'***********************************************************************
'***********************************************************************

1500  '********* SUBROUTINE TO WRITE A FILE ON DISK *************************************************

          OPEN FIL$ FOR OUTPUT AS #1
          PRINT #1, CITY.N$
          FOR I10 = 1 TO 7: FOR I11 = 1 TO 12: PRINT #1, BIDA(I10, I11); : NEXT I11: PRINT #1, : NEXT I10
          FOR I10 = 1 TO BIDA(1, 1): FOR I11 = 1 TO 8: FOR I12 = 1 TO 7:  PRINT #1, RODA(I10, I11, I12); : NEXT I12: PRINT #1, : NEXT I11
                                    FOR I11 = 1 TO 10: FOR I12 = 1 TO 20: PRINT #1, WIDA(I10, I11, I12); : NEXT I12: PRINT #1, : NEXT I11
                                    FOR I11 = 1 TO 10: FOR I12 = 1 TO 5:  PRINT #1, REFP(I10, I11, I12); : NEXT I12: PRINT #1, : NEXT I11
          NEXT I10
          CLOSE
          RETURN

1550  '*********** SUBROUTINE TO READ EXISTING FILE FROM DRIVE ****************************************************

          OPEN FIL$ FOR INPUT AS #1
          INPUT #1, CITY.N$
          FOR I10 = 1 TO 7: FOR I11 = 1 TO 12: INPUT #1, BIDA(I10, I11): NEXT I11  NEXT I10
          NSP = BIDA(1, 1):       GOSUB 155
          FOR I10 = 1 TO NSP: FOR I11 = 1 TO 8:  FOR I12 = 1 TO 7:  INPUT #1, RODA(I10, I11, I12): NEXT I12: NEXT I11
                                    FOR I11 = 1 TO 10: FOR I12 = 1 TO 20: INPUT #1, WIDA(I10, I11, I12): NEXT I12: NEXT I11
                                    FOR I11 = 1 TO 10: FOR I12 = 1 TO 5:  INPUT #1, REFP(I10, I11, I12): NEXT I12: NEXT I11
          NEXT I10
          CLOSE
          RETURN
```

```
******************************************************************************************
***************************** SUBPROGRAM FOR SCREEN PRESENTATION *****************************
******************************************************************************************

30000 CopyrightSS$ = "(C)Copyright 84,85 The Software Bottling Company Of New York"
30005 '   Variables Section For A-CITY
30010 '
30015 VLSS$(1) = CITY.N$: VLSS$(2) = STR$(WEATH(1, 1)): VLSS$(3) = STR$(WEATH(2, 1)):
30020 VLSS$(4) = STR$(WEATH(3, 1)): VLSS$(5) = STR$(WEATH(4, 1)):
30025 VLSS$(6) = STR$(WEATH(5, 1)): VLSS$(7) = STR$(WEATH(6, 1)):
30030 VLSS$(8) = STR$(WEATH(7, 1)): VLSS$(9) = STR$(WEATH(8, 1)):
30035 VLSS$(10) = STR$(WEATH(9, 1)): VLSS$(11) = STR$(WEATH(10, 1)):
30040 VLSS$(12) = STR$(WEATH(11, 1)): VLSS$(13) = STR$(WEATH(12, 1)):
30045 VLSS$(14) = STR$(WEATH(1, 3)): VLSS$(15) = STR$(WEATH(2, 3)):
30050 VLSS$(16) = STR$(WEATH(3, 3)): VLSS$(17) = STR$(WEATH(4, 3)):
30055 VLSS$(18) = STR$(WEATH(5, 3)): VLSS$(19) = STR$(WEATH(6, 3)):
30060 VLSS$(20) = STR$(WEATH(7, 3)): VLSS$(21) = STR$(WEATH(8, 3)):
30065 VLSS$(22) = STR$(WEATH(9, 3)): VLSS$(23) = STR$(WEATH(10, 3)):
30070 VLSS$(24) = STR$(WEATH(11, 3)): VLSS$(25) = STR$(WEATH(12, 3)):
30075 VLSS$(26) = STR$(WEATH(1, 7)): VLSS$(27) = STR$(WEATH(2, 7)):
30080 VLSS$(28) = STR$(WEATH(3, 7)): VLSS$(29) = STR$(WEATH(4, 7)):
30085 VLSS$(30) = STR$(WEATH(5, 7)): VLSS$(31) = STR$(WEATH(6, 7)):
30090 VLSS$(32) = STR$(WEATH(7, 7)): VLSS$(33) = STR$(WEATH(8, 7)):
30095 VLSS$(34) = STR$(WEATH(9, 7)): VLSS$(35) = STR$(WEATH(10, 7)):
30100 VLSS$(36) = STR$(WEATH(11, 7)): VLSS$(37) = STR$(WEATH(12, 7)):
30105 VLSS$(38) = STR$(WEATH(1, 6)): VLSS$(39) = STR$(WEATH(2, 6)):
30110 VLSS$(40) = STR$(WEATH(3, 6)): VLSS$(41) = STR$(WEATH(4, 6)):
30115 VLSS$(42) = STR$(WEATH(5, 6)): VLSS$(43) = STR$(WEATH(6, 6)):
30120 VLSS$(44) = STR$(WEATH(7, 6)): VLSS$(45) = STR$(WEATH(8, 6)):
30125 VLSS$(46) = STR$(WEATH(9, 6)): VLSS$(47) = STR$(WEATH(10, 6)):
30130 VLSS$(48) = STR$(WEATH(11, 6)): VLSS$(49) = STR$(WEATH(12, 6)):
30135 VLSS$(50) = STR$(WEATH(13, 1)): VLSS$(51) = STR$(WEATH(13, 2)):
30140 VLSS$(52) = STR$(WEATH(13, 4)): VLSS$(53) = STR$(WEATH(13, 3)):
30145 RETURN
30150 '
30155 '   Assign VLSS$ Array to the variables
30160 '
30165 CITY.N$ = VLSS$(1): WEATH(1, 1) = VAL(VLSS$(2)): WEATH(2, 1) = VAL(VLSS$(3)):
30170 WEATH(3, 1) = VAL(VLSS$(4)): WEATH(4, 1) = VAL(VLSS$(5)):
30175 WEATH(5, 1) = VAL(VLSS$(6)): WEATH(6, 1) = VAL(VLSS$(7)):
30180 WEATH(7, 1) = VAL(VLSS$(8)): WEATH(8, 1) = VAL(VLSS$(9)):
30185 WEATH(9, 1) = VAL(VLSS$(10)): WEATH(10, 1) = VAL(VLSS$(11)):
30190 WEATH(11, 1) = VAL(VLSS$(12)): WEATH(12, 1) = VAL(VLSS$(13)):
30195 WEATH(1, 3) = VAL(VLSS$(14)): WEATH(2, 3) = VAL(VLSS$(15)):
30200 WEATH(3, 3) = VAL(VLSS$(16)): WEATH(4, 3) = VAL(VLSS$(17)):
30205 WEATH(5, 3) = VAL(VLSS$(18)): WEATH(6, 3) = VAL(VLSS$(19)):
30210 WEATH(7, 3) = VAL(VLSS$(20)): WEATH(8, 3) = VAL(VLSS$(21)):
30215 WEATH(9, 3) = VAL(VLSS$(22)): WEATH(10, 3) = VAL(VLSS$(23)).
30220 WEATH(11, 3) = VAL(VLSS$(24)): WEATH(12, 3) = VAL(VLSS$(25)):
30225 WEATH(1, 7) = VAL(VLSS$(26)): WEATH(2, 7) = VAL(VLSS$(27)):
30230 WEATH(3, 7) = VAL(VLSS$(28)): WEATH(4, 7) = VAL(VLSS$(29)):
30235 WEATH(5, 7) = VAL(VLSS$(30)): WEATH(6, 7) = VAL(VLSS$(31)):
30240 WEATH(7, 7) = VAL(VLSS$(32)): WEATH(8, 7) = VAL(VLSS$(33)):
30245 WEATH(9, 7) = VAL(VLSS$(34)): WEATH(10, 7) = VAL(VLSS$(35)):
30250 WEATH(11, 7) = VAL(VLSS$(36)): WEATH(12, 7) = VAL(VLSS$(37)):
30255 WEATH(1, 6) = VAL(VLSS$(38)): WEATH(2, 6) = VAL(VLSS$(39)):
30260 WEATH(3, 6) = VAL(VLSS$(40)): WEATH(4, 6) = VAL(VLSS$(41)):
30265 WEATH(5, 6) = VAL(VLSS$(42)): WEATH(6, 6) = VAL(VLSS$(43)):
30270 WEATH(7, 6) = VAL(VLSS$(44)): WEATH(8, 6) = VAL(VLSS$(45)):
30275 WEATH(9, 6) = VAL(VLSS$(46)): WEATH(10, 6) = VAL(VLSS$(47)):
30280 WEATH(11, 6) = VAL(VLSS$(48)): WEATH(12, 6) = VAL(VLSS$(49)):
30285 WEATH(13, 1) = VAL(VLSS$(50)): WEATH(13, 2) = VAL(VLSS$(51)):
30290 WEATH(13, 4) = VAL(VLSS$(52)): WEATH(13, 3) = VAL(VLSS$(53)):
30295 RETURN
30300 '
30305 '   Section To Initialize Variables To Initial Values
30310 '
30315 CITY.N$ = "MONTREAL": WEATH(1, 1) = .4: WEATH(2, 1) = .4: WEATH(3, 1) = .4:
30320 WEATH(4, 1) = .4: WEATH(5, 1) = .4: WEATH(6, 1) = .4: WEATH(7, 1) = .4.
30325 WEATH(8, 1) = .4: WEATH(9, 1) = .4: WEATH(10, 1) = .4: WEATH(11, 1) = .4:
30330 WEATH(12, 1) = .4: WEATH(1, 3) = .2: WEATH(2, 3) = .2: WEATH(3, 3) = .2:
30335 WEATH(4, 3) = .2: WEATH(5, 3) = .2. WEATH(6, 3) = .2: WEATH(7, 3) = 2.
30340 WEATH(8, 3) = .2: WEATH(9, 3) = .2: WEATH(10, 3) = .2: WEATH(11, 3) = .2:
30345 WEATH(12, 3) = .2: WEATH(1, 7) = 870: WEATH(2, 7) = 767:
30350 WEATH(3, 7) = 653: WEATH(4, 7) = 380: WEATH(5, 7) = 176:
30355 WEATH(6, 7) = 38: WEATH(7, 7) = 5: WEATH(8, 7) = 24:
30360 WEATH(9, 7) = 92: WEATH(10, 7) = 289: WEATH(11, 7) = 490:
30365 WEATH(12, 7) = 773: WEATH(1, 6) = .39: WEATH(2, 6) = .49: WEATH(3, 6) = .55:
30370 WEATH(4, 6) = .51: WEATH(5, 6) = .51: WEATH(6, 6) = .5: WEATH(7, 6) = .53:
30375 WEATH(8, 6) = .53: WEATH(9, 6) = .47: WEATH(10, 6) = .43: WEATH(11, 6) = .32:
```

```
30380 WEATH(12, 6) = .33: WEATH(13, 1) = 46.5: WEATH(13, 2) = 74!: WEATH(13, 4) = 75:
30385 WEATH(13, 3) = .5:
30390 RETURN
30395 '
30400 '    **** List DATA statements & Print DISPLAY Only Variables ****
30405 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
30410 '
30415 DATA 58,2,10,"C","UUUUUUUUUU",,,1,6,0
30420 DATA 19,8,4,"N","#.##",0.00,1.00,0,6,1
30425 DATA 19,9,4,"N","#.##",0.00,1.00,0,6,1
30430 DATA 19,10,4,"N","#.##",0.00,1.00,0,6,1
30435 DATA 19,11,4,"N","#.##",0.00,1.00,0,6,1
30440 DATA 19,12,4,"N","#.##",0.00,1.00,0,6,1
30445 DATA 19,13,4,"N","#.##",0.00,1.00,0,6,1
30450 DATA 19,14,4,"N","#.##",0.00,1.00,0,6,1
30455 DATA 19,15,4,"N","#.##",0.00,1.00,0,6,1
30460 DATA 19,16,4,"N","#.##",0.00,1.00,0,6,1
30465 DATA 19,17,4,"N","#.##",0.00,1.00,0,6,1
30470 DATA 19,18,4,"N","#.##",0.00,1.00,0,6,1
30475 DATA 19,19,4,"N","#.##",0.00,1.00,0,6,1
30480 DATA 38,8,4,"N","#.##",0.00,1.00,0,6,1
30485 DATA 38,9,4,"N","#.##",0.00,1.00,0,6,1
30490 DATA 38,10,4,"N","#.##",0.00,1.00,0,6,1
30495 DATA 38,11,4,"N","#.##",0.00,1.00,0,6,1
30500 DATA 38,12,4,"N","#.##",0.00,1.00,0,6,1
30505 DATA 38,13,4,"N","#.##",0.00,1.00,0,6,1
30510 DATA 38,14,4,"N","#.##",0.00,1.00,0,6,1
30515 DATA 38,15,4,"N","#.##",0.00,1.00,0,6,1
30520 DATA 38,16,4,"N","#.##",0.00,1.00,0,6,1
30525 DATA 38,17,4,"N","#.##",0.00,1.00,0,6,1
30530 DATA 38,18,4,"N","#.##",0.00,1.00,0,6,1
30535 DATA 38,19,4,"N","#.##",0.00,1.00,0,6,1
30540 DATA 57,8,5,"N","####",1,9999,0,6,0
30545 DATA 57,9,5,"N","####",1,9999,0,6,0
30550 DATA 57,10,5,"N","####",1,9999,0,6,0
30555 DATA 57,11,5,"N","####",1,9999,0,6,0
30560 DATA 57,12,5,"N","####",1,9999,0,6,0
30565 DATA 57,13,5,"N","####",1,9999,0,6,0
30570 DATA 57,14,5,"N","####",1,9999,0,6,0
30575 DATA 57,15,5,"N","####",1,9999,0,6,0
30580 DATA 57,16,5,"N","####",1,9999,0,6,0
30585 DATA 57,17,5,"N","####",1,9999,0,6,0
30590 DATA 57,18,5,"N","####",1,9999,0,6,0
30595 DATA 57,19,5,"N","####",1,9999,0,6,0
30600 DATA 72,8,5,"N","#.##",0,1,0,6,1
30605 DATA 72,9,5,"N","#.##",0,1,0,6,1
30610 DATA 72,10,5,"N","#.##",0,1,0,6,1
30615 DATA 72,11,5,"N","#.##",0,1,0,6,1
30620 DATA 72,12,5,"N","#.##",0,1,0,6,1
30625 DATA 72,13,5,"N","#.##",0,1,0,6,1
30630 DATA 72,14,5,"N","#.##",0,1,0,6,1
30635 DATA 72,15,5,"N","#.##",0,1,0,6,1
30640 DATA 72,16,5,"N","#.##",0,1,0,6,1
30645 DATA 72,17,5,"N","#.##",0,1,0,6,1
30650 DATA 72,18,5,"N","#.##",0,1,0,6,1
30655 DATA 72,19,5,"N","#.##",0,1,0,6,1
30660 DATA 29,21,6,"N","###.##",-90.00,90.00,0,6,1
30665 DATA 68,21,7,"N","####.##",-180.00,180.00,0,6,1
30670 DATA 48,23,4,"N","####",-180,180,0,6,0
30675 DATA 74,23,4,"N","##.#",-1.0,10.0,0,6,1
30680 RETURN
30685 '
30690 '    Screen Display Initialization Statements
30695 '
30700 NUMFLDS.SS% = 53: FILNM.SS$ = "CITY.SCR":
30705 EXITCHR.SS$ = "|" + CHR$(127) + ""
30710 RESTORE 30400
30715 RETURN
30720 '
34005 '    Variables Section For C:BIDA
34010 '
34015 VLSS$(1) = FIL$: VLSS$(2) = STR$(BIDA(2, 1)): VLSS$(3) = STR$(BIDA(2, 2)):
34020 VLSS$(4) = STR$(BIDA(2, 3)): VLSS$(5) = STR$(BIDA(2, 4)):
34025 VLSS$(6) = STR$(BIDA(2, 5)): VLSS$(7) = STR$(BIDA(2, 6)):
34030 VLSS$(8) = STR$(BIDA(2, 7)): VLSS$(9) = STR$(BIDA(2, 8)):
34035 VLSS$(10) = STR$(BIDA(2, 9)): VLSS$(11) = STR$(BIDA(2, 10)):
34040 VLSS$(12) = STR$(BIDA(2, 11)): VLSS$(13) = STR$(BIDA(2, 12)):
34045 VLSS$(14) = STR$(BIDA(4, 1)): VLSS$(15) = STR$(BIDA(4, 2)):
34050 VLSS$(16) = STR$(BIDA(4, 3)): VLSS$(17) = STR$(BIDA(4, 4)):
34055 VLSS$(18) = STR$(BIDA(4, 5)): VLSS$(19) = STR$(BIDA(4, 6)):
```

```
34060 VL.SS$(20) = STR$(BIDA(4, 7)): VL.SS$(21) = STR$(BIDA(4, 8)):
34065 VL.SS$(22) = STR$(BIDA(4, 9)): VL.SS$(23) = STR$(BIDA(4, 10)):
34070 VL.SS$(24) = STR$(BIDA(4, 11)): VL.SS$(25) = STR$(BIDA(4, 12)):
34075 VL.SS$(26) = CITY.N$: VL.SS$(27) = STR$(BIDA(1, 1)):
34080 VL.SS$(28) = STR$(BIDA(7, 6)): VL.SS$(29) = STR$(BIDA(7, 4)):
34085 VL.SS$(30) = INS.W$: VL.SS$(31) = STR$(BIDA(7, 7)):
34090 VL.SS$(32) = STR$(BIDA(7, 8)):
34095 RETURN
34100 '
34105 '    Assign VL.SS$ Array to the variables
34110 '
34115 FIL$ = VL.SS$(1): BIDA(2, 1) = VAL(VL.SS$(2)): BIDA(2, 2) = VAL(VL.SS$(3)):
34120 BIDA(2, 3) = VAL(VL.SS$(4)): BIDA(2, 4) = VAL(VL.SS$(5)):
34125 BIDA(2, 5) = VAL(VL.SS$(6)): BIDA(2, 6) = VAL(VL.SS$(7)):
34130 BIDA(2, 7) = VAL(VL.SS$(8)): BIDA(2, 8) = VAL(VL.SS$(9)):
34135 BIDA(2, 9) = VAL(VL.SS$(10)): BIDA(2, 10) = VAL(VL.SS$(11)):
34140 BIDA(2, 11) = VAL(VL.SS$(12)): BIDA(2, 12) = VAL(VL.SS$(13)):
34145 BIDA(4, 1) = VAL(VL.SS$(14)): BIDA(4, 2) = VAL(VL.SS$(15)):
34150 BIDA(4, 3) = VAL(VL.SS$(16)): BIDA(4, 4) = VAL(VL.SS$(17)):
34155 BIDA(4, 5) = VAL(VL.SS$(18)): BIDA(4, 6) = VAL(VL.SS$(19)):
34160 BIDA(4, 7) = VAL(VL.SS$(20)): BIDA(4, 8) = VAL(VL.SS$(21)):
34165 BIDA(4, 9) = VAL(VL.SS$(22)): BIDA(4, 10) = VAL(VL.SS$(23)):
34170 BIDA(4, 11) = VAL(VL.SS$(24)): BIDA(4, 12) = VAL(VL.SS$(25)):
34175 CITY.N$ = VL.SS$(26): BIDA(1, 1) = VAL(VL.SS$(27)): BIDA(7, 6) = VAL(VL.SS$(28)):
34180 BIDA(7, 4) = VAL(VL.SS$(29)): INS.W$ = VL.SS$(30): BIDA(7, 7) = VAL(VL.SS$(31)):
34185 BIDA(7, 8) = VAL(VL.SS$(32)):
34190 RETURN
34195 '
34200 '    Section To Initialize Variables To Initial Values
34205 '
34210 FIL$ = "TEST": BIDA(2, 1) = 19: BIDA(2, 2) = 20: BIDA(2, 3) = 21: BIDA(2, 4) = 20:
34215 BIDA(2, 5) = 24: BIDA(2, 6) = 22: BIDA(2, 7) = 21: BIDA(2, 8) = 23: BIDA(2, 9) = 21:
34220 BIDA(2, 10) = 20: BIDA(2, 11) = 22: BIDA(2, 12) = 20: BIDA(4, 1) = 23!:
34225 BIDA(4, 2) = 23!: BIDA(4, 3) = 23!: BIDA(4, 4) = 23!: BIDA(4, 5) = 23!:
34230 BIDA(4, 6) = 25!: BIDA(4, 7) = 25!: BIDA(4, 8) = 25!: BIDA(4, 9) = 23!:
34235 BIDA(4, 10) = 23!: BIDA(4, 11) = 23!: BIDA(4, 12) = 23!: CITY.N$ = "MONTREAL":
34240 BIDA(1, 1) = 1: BIDA(7, 6) = 500: BIDA(7, 4) = 100: INS.W$ = "Y": BIDA(7, 7) = 8:
34245 BIDA(7, 8) = 17:
34250 RETURN
34255 '
34260 '    **** List DATA statements & Print DISPLAY Only Variables ****
34265 'Lin,Col,Len,Ficture,Low Ranga,High Range,Foreground,Background,# of Edit
34270 '
34275 DATA 58,2,6,"C","UUUUUU",,,1,6,0
34280 DATA 20,8,2,"N","##",0,31,0,6,0
34285 DATA 20,9,2,"N","##",0,31,0,6,0
34290 DATA 20,10,2,"N","##",0,31,0,6,0
34295 DATA 20,11,2,"N","##",0,31,0,6,0
34300 DATA 20,12,2,"N","##",0,31,0,6,0
34305 DATA 20,13,2,"N","##",0,31,0,6,0
34310 DATA 20,14,2,"N","##",0,31,0,6,0
34315 DATA 20,15,2,"N","##",0,31,0,6,0
34320 DATA 20,16,2,"N","##",0,31,0,6,0
34325 DATA 20,17,2,"N","##",0,31,0,6,0
34330 DATA 20,18,2,"N","##",0,31,0,6,0
34335 DATA 20,19,2,"N","##",0,31,0,6,0
34340 DATA 33,8,5,"N","###.#",15.0,30.0,0,6,1
34345 DATA 33,9,5,"N","###.#",15.0,30.0,0,6,1
34350 DATA 33,10,5,"N","###.#",15.0,30.0,0,6,1
34355 DATA 33,11,5,"N","###.#",15.0,30.0,0,6,1
34360 DATA 33,12,5,"N","###.#",15.0,30.0,0,6,1
34365 DATA 33,13,5,"N","###.#",15.0,30.0,0,6,1
34370 DATA 33,14,5,"N","###.#",15.0,30.0,0,6,1
34375 DATA 33,15,5,"N","###.#",15.0,30.0,0,6,1
34380 DATA 33,16,5,"N","###.#",15.0,30.0,0,6,1
34385 DATA 33,17,5,"N","###.#",15.0,30.0,0,6,1
34390 DATA 33,18,5,"N","###.#",15.0,30.0,0,6,1
34395 DATA 33,19,5,"N","###.#",15.0,30.0,0,6,1
34400 DATA 55,6,10,"C","UUUUUUUUUU",,,1,6,0
34405 DATA 59,11,2,"N","##",1,99,0,6,0
34410 DATA 58,15,4,"N","####",50,9999,0,6,0
34415 DATA 57,19,5,"N","#####",0,99999,0,6,0
34420 DATA 59,23,1,"Y","U",,,0,6,0
34425 DATA 30,21,2,"N","##",1,24,0,6,0
34430 DATA 30,23,2,"N","##",1,24,0,6,0
34435 RETURN
34440 '
34445 '    Screen Display Initialization Statements
34450 '
34455 NUMFLDS.SS% = 32: FILNM.SS$ = "BIDA.SCR":
```

```
34460 EXITCHR.SS$ = "I" + CHR$(127) + ""
34465 RESTORE 34260
34470 RETURN
31290 '
31295 '   Variables Section For A:RODA_1
31300 '
31305 VL.SS$(1) = INCL$:
31310 RETURN
31315 '
31320 '   Assign VL.SS$ Array to the variables
31325 '
31330 INCL$ = VL.SS$(1):
31335 RETURN
31340 '
31345 '   Section To Initialize Variables To Initial Values
31350 '
31355 INCL$ = "N":
31360 RETURN
31365 '
31370 '   **** List DATA statements & Print DISPLAY Only Variables ****
31375 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31380 '
31385 DATA 52,11,1,"Y","U",,,4,7,0
31390 COLOR 3, 6: LOCATE 2, 49: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
31395 COLOR 3, 6: LOCATE 2, 73: PRINT USING "##"; I;
31400 RETURN
31405 '
31410 '   Screen Display Initialization Statements
31415 '
31420 NUMFLDS.SS% = 1: FILNM.SS$ = "RODA_1.SCR":
31425 EXITCHR.SS$ = "I" + CHR$(127) + ""
31430 RESTORE 31370
31435 RETURN
31440 '
31445 '   Variables Section For A:RODA_2
31450 '
31455 VL.SS$(1) = STR$(RODA(I, 1, 1)): VL.SS$(2) = STR$(RODA(I, 2, 1)):
31460 VL.SS$(3) = STR$(RODA(I, 3, 2)):
31465 RETURN
31470 '
31475 '   Assign VL.SS$ Array to the variables
31480 '
31485 RODA(I, 1, 1) = VAL(VL.SS$(1)): RODA(I, 2, 1) = VAL(VL.SS$(2)):
31490 RODA(I, 3, 2) = VAL(VL.SS$(3)):
31495 RETURN
31500 '
31505 '   Section To Initialize Variables To Initial Values
31510 '
31515 RODA(I, 1, 1) = 1!: RODA(I, 2, 1) = 1!: RODA(I, 3, 2) = 1!:
31520 RETURN
31525 '
31530 '   **** List DATA statements & Print DISPLAY Only Variables ****
31535 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31540 '
31545 COLOR 3, 6: LOCATE 2, 49: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6),
31550 COLOR 3, 6: LOCATE 2, 73: PRINT USING "##"; I;
31555 DATA 21,18,5,"N","##.##",0.00,99.99,0,6,1
31560 DATA 38,13,5,"N","##.##",0.00,99.99,0,6,1
31565 DATA 50,12,5,"N","##.##",0.00,99.99,0,6,1
31570 COLOR 1, 6: LOCATE 7, 19: PRINT LEFT$(WALL.N$ + LEFT$(BLNK.SS$, 13 - LEN(WALL.N$)), 13);
31575 RETURN
31580 '
31585 '   Screen Display Initialization Statements
31590 '
31595 NUMFLDS.SS% = 3: FILNM.SS$ = "RODA_2.SCR":
31600 EXITCHR.SS$ = "I" + CHR$(127) + ""
31605 RESTORE 31530
31610 RETURN
31615 '
31620 '   Variables Section For A:RODA_3
31625 '
31630 VL.SS$(1) = STR$(RODA(I, 1, 3)): VL.SS$(2) = STR$(RODA(I, 1, 5)):
31635 VL.SS$(3) = STR$(RODA(I, 2, 5)): VL.SS$(4) = STR$(ROD   3, 5)):
31640 VL.SS$(5) = STR$(RODA(I, 4, 5)): VL.SS$(6) = IDENT.W$:
31645 VL.SS$(7) = STR$(RODA(I, 7, 5)): VL.SS$(8) = STR$(RODA(I, 1, 4)):
31650 VL.SS$(9) = STR$(RODA(I, 5, 4)): VL.SS$(10) = STR$(RODA(I, 6, 4)): VL.SS$(11) = STR$(RODA(I, 7, 7)):
31655 RETURN
31660 '
31665 '   Assign VL.SS$ Array to the variables
31670 '
```

```
31675 RODA(I, 1, 3) = VAL(VLSS$(1)): RODA(I, 1, 5) = VAL(VLSS$(2)):
31680 RODA(I, 2, 5) = VAL(VLSS$(3)): RODA(I, 3, 5) = VAL(VLSS$(4)):
31685 RODA(I, 4, 5) = VAL(VLSS$(5)): IDENT.W$ = VLSS$(6):
31690 RODA(I, 7, 5) = VAL(VLSS$(7)): RODA(I, 1, 4) = VAL(VLSS$(8)):
31695 RODA(I, 5, 4) = VAL(VLSS$(9)): RODA(I, 6, 4) = VAL(VLSS$(10)): RODA(I, 7, 7) = VAL(VLSS$(11)):
31700 RETURN
31705 '
31710 '   Section To Initialize Variables To Initial Values
31715 '
31720 RODA(I, 1, 3) = 0!: RODA(I, 1, 5) = 1: RODA(I, 2, 5) = 0: RODA(I, 3, 5) = 0:
31725 RODA(I, 4, 5) = 0: IDENT.W$ = "Y": RODA(I, 7, 5) = 1: RODA(I, 1, 4) = .7:
31730 RODA(I, 5, 4) = .7: RODA(I, 6, 4) = .7:  RODA(I, 7, 7) = .7:
31735 RETURN
31740 '
31745 '   **** List DATA statements & Print DISPLAY Only Variables ****
31750 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31755 '
31760 COLOR 3, 6: LOCATE 2, 52: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
31765 COLOR 3, 6: LOCATE 2, 75: PRINT USING "##"; I;
31770 DATA 10,5,5,"N","###.#",0.0,360.0,0,6,1
31775 DATA 73,8,2,"N","##",0,10,0,7,0
31780 DATA 73,10,2,"N","##",0,10,0,7,0
31785 DATA 73,12,2,"N","##",0,10,0,7,0
31790 DATA 73,14,2,"N","##",0,10,0,7,0
31795 DATA 73,18,1,"Y","U",,,4,7,0
31797 DATA 20,18,2,"N","##",1,10,0,7,0
31800 DATA 28,21,4,"N","#.##",0 10,1.00,0,6,1
31805 DATA 66,21,4,"N","#.##",0.10,1.00,0,6,1
31810 DATA 30,23,4,"N","# ##",0.10,1.00,0,6,1
31815 DATA 74,23,4,"N","#.##",0.10,1.00,0,6,1
31820 RETURN
31825 '
31830 '   Screen Display Initialization Statements
31835 '
31840 NUMFLDS.SS% = 11: FILNM.SS$ = "RODA_3.SCR":
31845 EXITCHR.SS$ = "!" + CHR$(127) + ""
31850 RESTORE 31745
31855 RETURN
31860 '
31865 '   Variables Section For A:RODA_I3
31870 '
31875 VLSS$(1) = STR$(RODA(I, 1, 3)): VLSS$(2) = STR$(RODA(I, 1, 6)):
31880 VLSS$(3) = STR$(RODA(I, 1, 5)): VLSS$(4) = IDENT.W$:
31885 VLSS$(5) = STR$(RODA(I, 1, 4)). VLSS$(6) = STR$(RODA(I, 5, 4)):
31890 VLSS$(7) = STR$(RODA(I, 6, 4)): VLSS$(8) = STR$(RODA(I, 7, 7)): VLSS$(9) = STR$(RODA(I, 7, 5)):
31895 RETURN
31900 '
31905 '   Assign VLSS$ Array to the variables
31910 '
31915 RODA(I, 1, 3) = VAL(VLSS$(1)): RODA(I, 1, 6) = VAL(VLSS$(2)):
31920 RODA(I, 1, 5) = VAL(VLSS$(3)): IDENT.W$ = VLSS$(4):
31925 RODA(I, 1, 4) = VAL(VLSS$(5)). RODA(I, 5, 4) = VAL(VLSS$(6)):
31930 RODA(I, 6, 4) = VAL(VLSS$(7)): RODA(I, 7, 7) = VAL(VLSS$(8)): RODA(I, 7, 5) = VAL(VLSS$(9)):
31935 RETURN
31940 '
31945 '   Section To Initialize Variables To Initial Values
31950 '
31955 RODA(I, 1, 3) = 0!: RODA(I, 1, 6) = 30!: RODA(I, 1, 5) = 1: IDENT.W$ = "Y":
31960 RODA(I, 1, 4) = .7: RODA(I, 5, 4) = .7: RODA(I, 6, 4) = .7: RODA(I, 7, 7) = .7: RODA(I, 7, 5) = 1:
31965 RETURN
31970 '
31975 '   **** List DATA statements & Print DISPLAY Only Variables ****
31980 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
31985 '
31990 COLOR 3, 6: LOCATE 2, 52: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
31995 COLOR 3, 6. LOCATE 2, 75: PRINT USING "##"; I;
32000 DATA 8,5,5,"N","###.#",0.0,360.0,0,6,1
32005 DATA 56,5,4,"N","##.#",0.0,90 0,0,6,1
32010 DATA 38,17,2,"N","##",1,10,0,6,0
32015 DATA 73,17,1,"Y","U",,,0,6,0
32020 DATA 28,19,4,"N","#.##",0 10,1.00,0,6,1
32025 DATA 66,19,4,"N","# ##",0.10,1.00,0,6,1
32030 DATA 30,21,4,"N","#.##",0 10,1.00,0,6,1
32035 DATA 74,21,4,"N","#.##",0.10,1.00,0,6,1
32037 DATA 59,23,2,"N","##",01,10,0,6,0
32040 RETURN
32045 '
32050 '   Screen Display Initialization Statements
32055 '
32060 NUMFLDS.SS% = 9: FILNM.SS$ = "RODA_I3.SCR":
```

```
32065 EXITCHR.SS$ = "I" + CHR$(127) + ""
32070 RESTORE 31975
32075 RETURN
32080 '
32085 '   Variables Section For A:RODA_4
32090 '
32095 VL.SS$(1) = STR$(RODA(I, 8, 7)): VL.SS$(2) = STR$(RODA(I, 8, 1)):
32100 VL.SS$(3) = STR$(RODA(I, 8, 2)): VL.SS$(4) = STR$(RODA(I, 8, 3)):
32105 VL.SS$(5) = STR$(RODA(I, 8, 4)): VL.SS$(6) = STR$(RODA(I, 8, 5)):
32110 VL.SS$(7) = STR$(RODA(I, 8, 6)):
32115 RETURN
32120 '
32125 '   Assign VL.SS$ Array to the variables
32130 '
32135 RODA(I, 8, 7) = VAL(VL.SS$(1)): RODA(I, 8, 1) = VAL(VL.SS$(2)):
32140 RODA(I, 8, 2) = VAL(VL.SS$(3)): RODA(I, 8, 3) = VAL(VL.SS$(4)):
32145 RODA(I, 8, 4) = VAL(VL.SS$(5)): RODA(I, 8, 5) = VAL(VL.SS$(6)):
32150 RODA(I, 8, 6) = VAL(VL.SS$(7)):
32155 RETURN
32160 '
32165 '   Section To Initialize Variables To Initial Values
32170 '
32175 RODA(I, 8, 7) = .5: RODA(I, 8, 1) = .75: RODA(I, 8, 2) = .75: RODA(I, 8, 3) = 80:
32180 RODA(I, 8, 4) = .75: RODA(I, 8, 5) = .9: RODA(I, 8, 6) = 3I:
32185 RETURN
32190 '
32195 '   **** List DATA statements & Print DISPLAY Only Variables ****
32200 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32205 '
32210 COLOR 3, 6: LOCATE 2, 49: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
32215 COLOR 3, 6: LOCATE 2, 73: PRINT USING "##"; I;
32220 DATA 22,9,3,"N","#.#",0,1,0,7,1
32225 DATA 38,9,4,"N","#.##",0.01,1.00,0,7,1
32230 DATA 60,9,4,"N","#.##",0.01,1.00,0,7,1
32235 DATA 11,19,4,"N","####",1,999,0,7,1
32240 DATA 29,19,4,"N","#.##",0.01,1.00,0,7,1
32245 DATA 47,19,4,"N","#.##",0.01,1.00,0,7,1
32250 DATA 65,19,5,"N","##.##",.05,99.99,0,7,1
32255 RETURN
32260 '
32265 '   Screen Display Initialization Statements
32270 '
32275 NUMFLDS.SS% = 7: FILNM.SS$ = "RODA_4.SCR":
32280 EXITCHR.SS$ = "I" + CHR$(127) + ""
32285 RESTORE 32195
32290 RETURN
32295 '
32300 '   Variables Section For A:WIDA_1
32305 '
32310 VL.SS$(1) = STR$(WIDA(I, I2, 2)): VL.SS$(2) = STR$(WIDA(I, I2, 4)):
32315 VL.SS$(3) = STR$(WIDA(I, I2, 3)): VL.SS$(4) = STR$(WIDA(I, I2, 5)):
32320 VL.SS$(5) = STR$(WIDA(I, I2, 7)): VL.SS$(6) = STR$(WIDA(I, I2, 8)):
32325 VL.SS$(7) = STR$(WIDA(I, I2, 20)): VL.SS$(8) = STR$(WIDA(I, I2, 18)):
32330 VL.SS$(9) = STR$(WIDA(I, I2, 17)): VL.SS$(10) = CHOICE1$: VL.SS$(11) = CHOICE2$:
32335 RETURN
32340 '
32345 '   Assign VL.SS$ Array to the variables
32350 '
32355 WIDA(I, I2, 2) = VAL(VL.SS$(1)): WIDA(I, I2, 4) = VAL(VL.SS$(2)):
32360 WIDA(I, I2, 3) = VAL(VL.SS$(3)): WIDA(I, I2, 5) = VAL(VL.SS$(4)):
32365 WIDA(I, I2, 7) = VAL(VL.SS$(5)): WIDA(I, I2, 8) = VAL(VL.SS$(6)):
32370 WIDA(I, I2, 20) = VAL(VL.SS$(7)): WIDA(I, I2, 18) = VAL(VL.SS$(8)):
32375 WIDA(I, I2, 17) = VAL(VL.SS$(9)): CHOICE1$ = VL.SS$(10): CHOICE2$ = VL.SS$(11):
32380 RETURN
32385 '
32390 '   Section To Initialize Variables To Initial Values
32395 '
32400 WIDA(I, I2, 2) = 1I: WIDA(I, I2, 4) = 1I: WIDA(I, I2, 3) = 1I:
32405 WIDA(I, I2, 5) = 1I: WIDA(I, I2, 7) = .8: WIDA(I, I2, 8) = .8:
32410 WIDA(I, I2, 20) = 2.4: WIDA(I, I2, 18) = .6: WIDA(I, I2, 17) = 16: CHOICE1$ = "N":
32415 CHOICE2$ = "N":
32420 RETURN
32425 '
32430 '   **** List DATA statements & Print DISPLAY Only Variables ****
32435 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32440 '
32445 COLOR 3, 6: LOCATE 2, 52: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
32450 COLOR 3, 6: LOCATE 2, 75: PRINT USING "##"; I;
32455 DATA 12,15,5,"N","##.##",0.00,99.99,7,6,1
32460 DATA 20,15,5,"N","##.##",0.00,99.99,7,6,1
```

```
32465 DATA 29,14,5,"N","##.##",0.00,99.99,7,6,1
32470 DATA 29,11,5,"N","##.##",0.00,99.99,7,6,1
32475 COLOR 3, 6: LOCATE 4, 67: PRINT USING "##"; I4;
32480 DATA 33,21,4,"N","#.##",0.01,1.00,0,6,1
32485 DATA 73,21,4,"N","#.##",0.01,1.00,0,6,1
32490 DATA 23,23,3,"N","#.#",0,9.9,0,6,1
32495 DATA 51,23,4,"N","##.#",0.1,99.0,0,6,1
32500 DATA 76,23,2,"N","##",4,32,0,6,0
32505 DATA 72,9,1,"Y","U",,,0,7,0
32510 DATA 70,16,1,"Y","U",,,0,7,0
32515 COLOR 1, 6: LOCATE 4, 14: PRINT LEFT$(WALL.N$ + LEFT$(BLNK.SS$, 13 - LEN(WALL.N$)), 13);
32520 RETURN
32525 '
32530 '    Screen Display Initialization Statements
32535 '
32540 NUMFLDS.SS% = 11: FILNM.SS$ = "WIDA_1.SCR":
32545 EXITCHR.SS$ = "I" + CHR$(127) + ""
32550 RESTORE 32430
32555 RETURN
32560 '
32565 '    Variables Section For A:WIDA_2
32570 '
32575 VL.SS$(1) = STR$(WIDA(I, I2, 11)): VL.SS$(2) = STR$(WIDA(I, I2, 9)):
32580 VL.SS$(3) = STR$(WIDA(I, I2, 10)): VL.SS$(4) = STR$(WIDA(I, I2, 12)):
32585 RETURN
32590 '
32595 '    Assign VL.SS$ Array to the variables
32600 '
32605 WIDA(I, I2, 11) = VAL(VL.SS$(1)): WIDA(I, I2, 9) = VAL(VL.SS$(2)):
32610 WIDA(I, I2, 10) = VAL(VL.SS$(3)): WIDA(I, I2, 12) = VAL(VL.SS$(4)):
32615 RETURN
32620 '
32625 '    Section To Initialize Variables To Initial Values
32630 '
32635 WIDA(I, I2, 11) = 0!: WIDA(I, I2, 9) = 0!: WIDA(I, I2, 10) = 0!:
32640 WIDA(I, I2, 12) = 0!:
32645 RETURN
32650 '
32655 '    **** List DATA statements & Print DISPLAY Only Variables ****
32660 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32665 '
32670 COLOR 3, 6: LOCATE 2, 52: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
32675 COLOR 3, 6: LOCATE 2, 75: PRINT USING "##"; I;
32680 COLOR 3, 6: LOCATE 4, 67: PRINT USING "##"; I4;
32685 COLOR 1, 6: LOCATE 4, 15: PRINT LEFT$(WALL.N$ + LEFT$(BLNK.SS$, 13 - LEN(WALL.N$)), 13);
32690 DATA 19,19,5,"N","##.##",-9.00,9.99,7,6,1
32695 DATA 51,13,4,"N","#.##",0.00,9.99,7,6,1
32700 DATA 61,10,4,"N","#.##",0.00,9.99,7,6,1
32705 DATA 69,13,4,"N","#.##",0.00,9.99,7,6,1
32710 RETURN
32715 '
32720 '    Screen Display Initialization Statements
32725 '
32730 NUMFLDS.SS% = 4: FILNM.SS$ = "WIDA_2.SCR":
32735 EXITCHR.SS$ = "I" + CHR$(127) + ""
32740 RESTORE 32655
32745 RETURN
32750 '
32755 '    Variables Section For A:WIDA_3
32760 '
32765 VL.SS$(1) = STR$(WIDA(I, I2, 14)): VL.SS$(2) = STR$(WIDA(I, I2, 16)):
32770 VL.SS$(3) = STR$(WIDA(I, I2, 15)): VL.SS$(4) = STR$(WIDA(I, I2, 13)):
32775 RETURN
32780 '
32785 '    Assign VL.SS$ Array to the variables
32790 '
32795 WIDA(I, I2, 14) = VAL(VL.SS$(1)): WIDA(I, I2, 16) = VAL(VL.SS$(2)):
32800 WIDA(I, I2, 15) = VAL(VL.SS$(3)): WIDA(I, I2, 13) = VAL(VL.SS$(4)):
32805 RETURN
32810 '
32815 '    Section To Initialize Variables To Initial Values
32820 '
32825 WIDA(I, I2, 14) = 0!: WIDA(I, I2, 16) = 0!: WIDA(I, I2, 15) = 0!:
32830 WIDA(I, I2, 13) = 0!:
32835 RETURN
32840 '
32845 '    **** List DATA statements & Print DISPLAY Only Variables ****
32850 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
32855 '
32860 COLOR 3, 6: LOCATE 2, 52: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
```

```
32865 COLOR 3, 6: LOCATE 2, 75: PRINT USING "##"; I;
32870 COLOR 3, 6: LOCATE 4, 67: PRINT USING "##"; I4;
32875 COLOR 1, 6: LOCATE 4, 15: PRINT LEFT$(WALL.N$ + LEFT$(BLNK.SS$, 13 - LEN(WALL.N$)), 13);
32880 DATA 21,19,4,"N","#.##",0.00,9.99,7,6,1
32885 DATA 49,13,5,"N","##.##",-9.99,99.99,7,6,1
32890 DATA 46,18,5,"N","##.##",-9.99,99.99,7,6,1
32895 DATA 60,10,4,"N","#.##",0.00,9.99,7,6,1
32900 RETURN
32905 '
32910 '    Screen Display Initialization Statements
32915 '
32920 NUMFLDS.SS% = 4: FILNM.SS$ = "WIDA_3.SCR":
32925 EXITCHR.SS$ = "I" + CHR$(127) + ""
32930 RESTORE 32845
32935 RETURN
32940 '
32945 '    Variables Section For A:REFP
32950 '
32955 VL.SS$(1) = STR$(REFP(I, I3, 1)): VL.SS$(2) = STR$(REFP(I, I3, 2)):
32960 VL.SS$(3) = STR$(REFP(I, I3, 3)): VL.SS$(4) = STR$(REFP(I, I3, 5)):
32970 RETURN
32975 '
32980 '    Assign VL.SS$ Array to the variables
32985 '
32990 REFP(I, I3, 1) = VAL(VL.SS$(1)): REFP(I, I3, 2) = VAL(VL.SS$(2)):
32995 REFP(I, I3, 3) = VAL(VL.SS$(3)): REFP(I, I3, 5) = VAL(VL.SS$(4)):
33005 RETURN
33010 '
33015 '    Section To Initialize Variables To Initial Values
33020 '
33025 REFP(I, I3, 1) = 1!: REFP(I, I3, 2) = 1!: REFP(I, I3, 3) = 1!:
33030 REFP(I, I3, 5) = 1!:
33035 RETURN
33040 '
33045 '    **** List DATA statements & Print DISPLAY Only Variables ****
33050 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
33055 '
33060 COLOR 3, 6: LOCATE 2, 52: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
33065 COLOR 3, 6: LOCATE 2, 75: PRINT USING "##"; I;
33070 DATA 18,18,5,"N","##.##",0.00,99.99,0,6,1
33075 DATA 38,13,5,"N","##.##",0.00,99.99,0,6,1
33080 DATA 49,14,5,"N","##.##",0.00,99.99,0,6,1
33085 COLOR 3, 6: LOCATE 4, 47: PRINT USING "##"; I3;
33090 DATA 59,23,4,"N","##.#",0.0,99.9,0,6,1
33100 COLOR 4, 6: LOCATE 7, 19: PRINT LEFT$(WALL.N$ + LEFT$(BLNK.SS$, 13 - LEN(WALL.N$)), 13);
33105 RETURN
33110 '
33115 '    Screen Display Initialization Statements
33120 '
33125 NUMFLDS.SS% = 4: FILNM.SS$ = "REFP.SCR":
33130 EXITCHR.SS$ = "I" + CHR$(127) + ""
33135 RESTORE 33045
33140 RETURN
33145 '
33150 '    Variables Section For A:ENDS
33155 '
33160 '
33165 RETURN
33170 '
33175 '    Assign VL.SS$ Array to the variables
33180 '
33185 '
33190 RETURN
33195 '
33200 '    Section To Initialize Variables To Initial Values
33205 '
33210 '
33215 RETURN
33220 '
33225 '    **** List DATA statements & Print DISPLAY Only Variables ****
33230 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
33235 '
33240 RETURN
33245 '
33250 '    Screen Display Initialization Statements
33255 '
33260 NUMFLDS.SS% = 0: FILNM.SS$ = "ENDS.SCR":
33265 EXITCHR.SS$ = "I" + CHR$(127) + ""
33270 RESTORE 33225
33275 RETURN
```

```
34480 '   Variables Section For C:BIDA_1
34485 '
34490 VLSS$(1) = STR$(BIDA(7, 1)): VLSS$(2) = STR$(BIDA(7, 2)):
34495 VLSS$(3) = STR$(BIDA(7, 3)): VLSS$(4) = STR$(BIDA(7, 5)):
34500 RETURN
34505 '
34510 '   Assign VLSS$ Array to the variables
34515 '
34520 BIDA(7, 1) = VAL(VLSS$(1)): BIDA(7, 2) = VAL(VLSS$(2)):
34525 BIDA(7, 3) = VAL(VLSS$(3)): BIDA(7, 5) = VAL(VLSS$(4)):
34530 RETURN
34535 '
34540 '   Section To Initialize Variables To Initial Values
34545 '
34550 BIDA(7, 1) = 900: BIDA(7, 2) = 500: BIDA(7, 3) = 100: BIDA(7, 5) = 300:
34555 RETURN
34560 '
34565 '   **** List DATA statements & Print DISPLAY Only Variables ****
34570 'Lin,Col,Len,Picture,Low Range,High Range,Foreground,Background,# of Edit
34575 '
34580 COLOR 3, 6: LOCATE 2, 62: PRINT LEFT$(FIL$ + LEFT$(BLNK.SS$, 6 - LEN(FIL$)), 6);
34585 DATA 9,19,6,"N","######",0,999999,0,7,0
34590 DATA 29,19,5,"N","#####",0,99999,0,7,0
34595 DATA 47,19,4,"N","####",0,9999,0,7,0
34600 DATA 65,19,5,"N","#####",0,99999,0,7,0
34605 RETURN
34610 '
34615 '   Screen Display Initialization Statements
34620 '
34625 NUMFLDS.SS% = 4: FILNM.SS$ = "BIDA_1.SCR":
34630 EXITCHR.SS$ = "I" + CHR$(127) + ""
34635 RESTORE 34565
34640 RETURN
60000 '
60010 ' ┌──────────────────────────────────────────────────────────┐
60020 ' | START OF SCREEN SCULPTOR FULL SCREEN EDITING ROUTINE  |
60030 ' └──────────────────────────────────────────────────────────┘
60050 '            ┌────────────────────────────┐
60060 '            |  Main body of subroutine  |
60070 '            └────────────────────────────┘
60080 '
60085 IF SAME.SS% THEN 60200  'To return to the SAME screen with SAME values
60090 IF SCR.SS% = SCRLST.SS% THEN 60140' If same screen as last, don't reload
60100                           ' Get screen setup parameters
60105   ON SCR.SS% GOSUB 30690, 34445, 31410, 31585, 31830, 32050, 32265, 32530, 32720, 32910, 33115, 33250, 34615
60110   GOSUB 60550             ' Read field data for this screen
60120   OUT &H3D8, &H1          ' Turn off screen display
60125 ' CALL QUBLOAD(FILNM.SS$)     '<- For QuickBASIC, See READ.ME file
60130   DEF SEG = SCRNSEG.SS%: BLOAD FILNM.SS$, 0: DEF SEG ' Load screen picture
60135                           ' Set initial values
60140   IF INIT.SS% THEN ON SCR.SS% GOSUB 30305, 34200, 31345, 31505, 31710, 31945, 32165, 32390, 32625, 32815, 33015, 33200, 34540
60145                           ' Assign current values to screen array
60155   ON SCR.SS% GOSUB 30005, 34005, 31295, 31445, 31620, 31865, 32085, 32300, 32565, 32755, 32945, 33150, 34480
60160   OUT &H3D8, &H29         ' Turn on screen display
60170   GOSUB 60590             ' Pad fields with blanks and display
60175                           ' Display initial DISPLAY variables
60180   ON SCR.SS% GOSUB 30400, 34260, 31370, 31530, 31745, 31975, 32195, 32430, 32655, 32845, 33045, 33225, 34565
60185   OUT &H3D8, &H29         ' Turn on screen display
60190 '
60195 F.SS% = 1: SCRLST.SS% = SCR.SS%
60200 COLOR 7, 0: LOCATE 25, 1: PRINT BLNK.SS$; 'Clr msg from prior screen
60205 IF NUMFLDS.SS% = 0 THEN RETURN      'Exit if no fields on screen
60210 LOCATE , , , 0, 1              'Make cursor size large
60215 EXSCR.SS% = 0                     'Initialize Flag For Screen Exit
60220 WHILE NOT EXSCR.SS%       'Loop on each field(until Exit Flag is set)
60250   GOSUB 60740                  'Accept input data for this field
60260   '
60265   ' The above subroutine accepts data for a single field.  The          program will return to this spot after the cursor exits any field on the input
                  screen.
60270   '
60275   ' If you want to do any special input field testing, this is
60280   ' a good place to do it.
60285   '
60290   ' The following variables are passed back for your use:
60295   ' F.SS% is next field to be edited
60300   ' FLDLST.SS% is last field edited
60305   ' LASTCHR.SS$ is last keyboard character entered
60310   ' VLSS$(n) contains the current value of field n.
60315   '
60320 WEND
```

```
60360 '
60370 '
60390 'COLOR 7, 0: LOCATE 25, 1: PRINT BLNK.SS$; : LOCATE 25, 15: PRINT "... Please WAIT A Moment While Checking Fields ...";
60400 FOR F.SS% = 1 TO NUMFLDS.SS%   ' Test Each Field Before Leaving Screen
60410     GOSUB 62500                ' Check contents of this field
60420     IF ERR.MSG% = -1 THEN EXSCR.SS% = 0: GOTO 60220'Error Detected
60430 NEXT F.SS%
60440 F.SS% = FLDLST.SS%                  'Reset Field indicator
60450 '
60455                              'Assign new field values
60460 ON SCR.SS% GOSUB 30155, 34105, 31320, 31475, 31665, 31905, 32125, 32345, 32595, 32785, 32980, 33175, 34510
60470 RETURN                            'Exit this subroutine
60480 '
60490 '********************************************************************************************************
```

THE CALCULATIONS MODULE

Main \ Diffuse Sky Illuminance Routine

```
DECLARE SUB SCREENN ()
************************ INITIALIZATION ****************************************************************************

        DIM SHARED BIDA(7, 12), FILNM.SS$
        PI = 3.141592653#:     I4.M% = 0:
        OPEN "ROUTF.AAA" FOR INPUT AS #1: INPUT #1, ROUT.N%: CLOSE
        ON ROUT.N% GOTO 30100, 30150, 30200, 30250, 30300, 30350, 30400, 30450, 30500
155     DIM SHARED RODA(NSP, 8, 16), WIDA(NSP, 10, 20), REFP(NSP, 10, 6), WEATH(13, 7), REFIL(NSP, 10, 24, 3), REFS(NSP, 18, 3), REFW(NSP,
        18, 3), REFT(NSP, 18, 3)
        DIM SHARED EKH1(24, 3), EDH1(24, 3), EDH(3)
        DIM SHARED TENS(12), WASE(NSP, 12), RADT(NSP, 72), TMSL(NSP, 12), TMHR(NSP, 12), TEMP(72, 12), QW(72), QT(72), QS(72), TE(72),
        QL(72),QC(72), QG(72), GTT(NSP, 4, 72)
        DIM LCONS(24, 3), RBHLT(12), RBAUXIL(12), RBEREQ(12), BEREQ(12), BAUXIL(12), BHLDT(12), QS1(12), BHFLIGHT(12), QG1(72), RQC(72),
        TRADT(72),        BTMSL(12), BMLCONS(12)
        RETURN

********************************************************************************************************************

5200  ******** SUBROUTINE TO CALCULATE INCIDENT ANGLE ON A SURFACE ***********************************************
        '(req:- SLOP, SALT)

        RA9 = COS(SALT) * COS(AZMD) * SIN(SLOP) + SIN(SALT) * COS(SLOP)
        INANG = ATN(SQR(1 - RA9 ^ 2) / RA9)
        IF INANG < 0 THEN INANG = INANG + PI
        RETURN

********************************************************************************************************************
************************ SUN POSITION CALCULATIONS *****************************************************************
********************************************************************************************************************

10100 ************* SUBROUTINE TO CALCULATE SOLAR TIME (ST) *********************************************************
        'Req.  JULN, STT(decimal hours), STM, LONGT(radians)

        ET = .17 * SIN(4 * PI * (JULN - 80) / 373) - .129 * SIN(2 * PI * (JULN - 8) / 355)
        ST = STT + ET + 12 * (STM - LONGT) / PI
        RETURN

10150 ** SUBROUTINE TO CALCULATE SOLAR ALTITUDE (SALT) & SOLAR AZIMUTH (SAZMN) & ZENITH ANGLE (ZANG) **************
        'Req    JULN, ST(decimal hours), LAT(radians)

        RA3 = 0:   'to determine whether the sun located in east or west side
        SDEC = .4093 * SIN(2 * PI * (JULN - 81) / 368)
        RA1 = SIN(LAT) * SIN(SDEC) - COS(LAT) * COS(SDEC) * COS(PI * ST / 12)
        SALT = ATN(RA1 / SQR(1 - RA1 ^ 2)):       ZANG = (PI / 2) - SALT
        IF SALT < 0 THEN RETURN
        IF INT(SALT * 100) = 157 THEN SAZMN = 0: SAZMS = 0: RETURN
        IF ST = 12 THEN SAZMN = PI: SAZMS = 0: RETURN
        IF ST > 12 THEN ST = 24 - ST:         RA3 = 1
        RA2 = (COS(LAT) * SIN(SDEC) + SIN(LAT) * COS(SDEC) * COS(PI * ST / 12)) / COS(SALT)
        SAZMN = ATN(SQR(1 - RA2 ^ 2) / RA2)
        IF SAZMN < 0 THEN SAZMN = SAZMN + PI
        IF RA3 = 1 THEN SAZMN = 2 * PI - SAZMN:  ST = 24 - ST
        SAZMS = SAZMN - PI
        RETURN

10250 ***** SUBROUTINE TO CALCULATE HOUR ANGLE (HANG) & SUNSET HOUR ANGLE (SSHANG) ********************************

        HANG = (ST - 12) * 15 * PI / 180:
        RA4 = (-1) * TAN(LAT) * TAN(SDEC)
        SSHANG = ATN(SQR(1 - RA4 ^ 2) / RA4)
        IF SSHANG < 0 THEN SSHANG = SSHANG + PI
        RETURN
```

```
************************************************************************************************************
**************************** DAYLIGHTING CALCULATIONS ******************************************************
************************************************************************************************************

11000 '********** SUBROUTINE TO CALCULATE HORIZONTAL ILLUMINANCE FROM THE SKY (EKH) ****************************
      'req:- sky condition SKCO  1=clear  2=partialy clouded  3=clouded , SALT

      IF SKCO = 1 THEN RA30 = .8:   RA31 = 15.5:   RA32 = .5
      IF SKCO = 2 THEN RA30 = .3:   RA31 = 45:   RA32 = 1
      IF SKCO = 3 THEN RA30 = .3:   RA31 = 21:   RA32 = 1
      EKH = RA30 + RA31 * (SIN(SALT)) ^ RA32:        '(** Kilo Lux **)
      EKH1(STT, SKCO) = INT(EKH * 100) / 100
      RETURN

11450 '***************** SUBROUTINE TO CALCULATE  SKY ILLUMINANCE ON A TILTED WINDOW ****************************
      'req:

      IF I4.M% = 0 THEN SLOP = PI / 2 - WIDA(I, I2, 6):
      IF I4.M% = 1 THEN AZMD = ABS(SAZMS - AZM) ELSE AZMD = ABS(SAZMS - WIDA(I, I2, 10))
      IF AZMD > PI THEN AZMD = (2 * PI) - AZMD
      SSALT = CINT(SALT * 18 / PI): SSLOP = CINT(SLOP * 18 / PI): SAZMD = CINT(AZMD * 9 / PI)
      Filenum = FREEFILE
      OPEN "ILLUM.AHM" FOR RANDOM AS #Filenum LEN = 32
      FIELD #Filenum, 5 AS A$
      CODE% = (SKCO * 1000) + (SSALT * 100) + (SSLOP * 10) + (SAZMD)
      GET #Filenum, CODE%
      LWIND = CVS(A$)
      IF I4.M% = 1 THEN RETURN
      LWIND = LWIND * WIDA(I, I2, 7) * WIDA(I, I2, 8)
      RETURN

13100 '***************** SUBROUTINE TO PREPARE WALLS # ACCORDING TO WINDOW UNDER CONSIDERATION (W1, W2, W3, W4) ******************
      '                       WHERE W1 IS THE WINDOW WALL

      IF WIDA(I, I2, 1) = 1 THEN W1 = 1:   W2 = 2:   W3 = 3:   W4 = 4
      IF WIDA(I, I2, 1) = 2 THEN W1 = 2:   W2 = 3:   W3 = 4:   W4 = 1
      IF WIDA(I, I2, 1) = 3 THEN W1 = 3:   W2 = 4:   W3 = 1:   W4 = 2
      IF WIDA(I, I2, 1) = 4 THEN W1 = 4:   W2 = 1:   W3 = 2:   W4 = 3
      RETURN

13200 '***************** SUBROUTINT TO PREPARE WINDOW VARIABLES ****************************************************
      'req. I, I2, W1, W2, W3, W4 (WALLS #)

      WALT = WIDA(I, I2, 6):      Y = WIDA(I, I2, 3):    F = WIDA(I, I2, 5):    ANG = ATN(WIDA(I, I2, 10) / (.5 * WIDA(I, I2, 5)))
      M = WIDA(I, I2, 4):          X = WIDA(I, I2, 2):    WID = RODA(I, W2, 1)
      LENGTH = RODA(I, W1, 1):      HIGHT = RODA(I, W2, 2)
      RETURN

13400 '***************** SUBROUTINE TO CALCULATE BEAM ILLUMINANCE ON A SURFACE   ( K.LUX ) **************************
      'req: JULN, SALT, SKCO, INANG

      IF SKCO = 3 THEN EDH = 0: EDV = 0: RETURN
      IF SKCO = 1 THEN C = .21 ELSE C = .8
      EXTS = 127.5 * (1 + .034 * COS(2 * PI * (JULN - 2) / 365))
      EDN = EXTS * EXP(-1 * C / SIN(SALT))
      EDH = EDN * SIN(SALT):       EDH(STT) = EDH
      IF AZMD > (PI / 2) THEN EDV = 0 ELSE EDV = EDN * COS(INANG)
      RETURN

13550 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE IN A POINT FROM TILTED SOURCE ************************

      AAA = SQR(H ^ 2 + Q ^ 2 - 2 * Q * H * COS(SLOP1))
      BBB = SQR(W ^ 2 + Q ^ 2 * (SIN(SLOP1)) ^ 2)
      RA1 = ATN((H - Q * COS(SLOP1)) / BBB) + ATN((Q * COS(SLOP1)) / BBB)
      ETILT = (L / 2) * (ATN(W / Q) + (H * COS(SLOP1) - Q) * ATN(W / AAA) / AAA + (W * COS(SLOP1) * RA1 / BBB))
      RETURN

13600 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE IN A POINT FROM PARALLEL SOURCE ***********************
      'req: Q, H, W (dimensions),  L (luminance)

      RA1 = SQR(H ^ 2 + Q ^ 2)
      RA2 = SQR(W ^ 2 + Q ^ 2)
      EPAR = (L / 2) * ((H / RA1) * ATN(W / RA1) + (W / RA2) * ATN(H / RA2))
      RETURN

13650 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE IN A POINT FROM PERPENDICULAR SOURCE ******************
      'req: Q, H, W (dimensions),  L (luminance)

      RA1 = SQR(H ^ 2 + Q ^ 2)
      EPRN = (L / 2) * (ATN(W / Q) - (Q / RA1) * ATN(W / RA1))
```

```
        RETURN

13700 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13600 AND GET FINAL RESULTS **************************
        'req: F, M, N, S, R, U (as in drawings)    { M+N < U or U < N }

        Q = F:    IF U > M + N THEN H = U - N ELSE H = M + N - U:  W = S + R:     GOSUB 13600:  LUM1 = EPAR:   '(ABCD)
        Q = F:    IF U > M + N THEN H = U - N ELSE H = M + N - U:  W = S:         GOSUB 13600:  LUM2 = EPAR:   '(BC)
        Q = F:    IF U > M + N THEN H = U - M - N ELSE H = N - U
                    W = S + R:   GOSUB 13600:  LUM3 = EPAR:     '(CD)
        Q = F:    IF U > M + N THEN H = U - M - N ELSE H = N - U
                    W = S:       GOSUB 13600:  LUM4 = EPAR:     '(C)
        ELL = LUM1 - LUM2 - LUM3 + LUM4
        RETURN

13710 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13600 AND GET FINAL RESULTS **************************
        'req: F, M, N, K, J, U (as in drawings), L (Luminance)    { M+N < U or U < N }

        Q = F:    IF U > M + N THEN H = U - N ELSE H = M + N - U:
        W = K:    GOSUB 13600:  LUM1 = EPAR:    '(AD)
        Q = F:    IF U > M + N THEN H = U - N ELSE H = M + N - U:
        W = J:    GOSUB 13600:  LUM2 = EPAR:    '(A1D1)
        Q = F:    IF U > M + N THEN H = U - M - N ELSE H = N - U
                    W = K:    GOSUB 13600:  LUM3 = EPAR:  '(D)
        Q = F:    IF U > M + N THEN H = U - M - N ELSE H = N - U
                    W = J:    GOSUB 13600:  LUM4 = EPAR:  '(D1)
        ELL = LUM1 + LUM2 - LUM3 - LUM4
        RETURN

13720 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13600 AND GET FINAL RESULTS **************************
        'req: F, M, N, S, R (as in drawings)    { N < U < M+N }

        Q = F:    H = M + N - U:    W = S + R:    GOSUB 13600:  LUM1 = EPAR:    '(AB)
        Q = F:    H = U - N:        W = S + R:    GOSUB 13600:  LUM2 = EPAR:    '(CD)
        Q = F:    H = M + N - U:    W = S:        GOSUB 13600:  LUM3 = EPAR:    '(B)
        Q = F:    H = U - N:        W = S:        GOSUB 13600:  LUM4 = EPAR:    '(C)
        ELL = LUM1 + LUM2 - LUM3 - LUM4
        RETURN

13730 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13600 AND GET FINAL RESULTS **************************
        'req: F, M, N, K, J (as in drawings), L (Luminance)    { N < U < M+N }

        Q = F:    H = M + N - U:  W = K:    GOSUB 13600:  LUM1 = EPAR:    '(A)
        Q = F:    H = M + N - U:  W = J:    GOSUB 13600:  LUM2 = EPAR:    '(B)
        Q = F:    H = U - N:      W = J:    GOSUB 13600:  LUM3 = EPAR:    '(C)
        Q = F:    H = U - N:      W = K:    GOSUB 13600:  LUM4 = EPAR:    '(D)
        ELL = LUM1 + LUM2 + LUM3 + LUM4
        RETURN

13750 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13650 AND GET FINAL RESULTS **************************
        'req: F, M, N, S, R (as in drawings)

        IF N < 0 THEN N = 0
        Q = F:    H = M + N:    W = S + R:    GOSUB 13650:  LUM1 = EPRN:    '(ABCD)
        Q = F:    H = M + N:    W = S:        GOSUB 13650:  LUM2 = EPRN.    '(BC)
        Q = F:    H = N:        W = S + R:    GOSUB 13650:  LUM3 = EPRN:    '(CD)
        Q = F:    H = N:        W = S:        GOSUB 13650:  LUM4 = EPRN:    '(C)
        ELL = LUM1 - LUM2 - LUM3 + LUM4
        RETURN

13760 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13650 AND GET FINAL RESULTS **************************
        'req: F, M, N, K, J (as in drawings), L (Luminance)

        IF N < 0 THEN N = 0
        Q = F:    H = M + N:    W = K:    GOSUB 13650:  LUM1 = EPRN:    '(AD)
        Q = F:    H = M + N:    W = J:    GOSUB 13650:  LUM2 = EPRN:    '(A1D1)
        Q = F:    H = N:        W = K:    GOSUB 13650:  LUM3 = EPRN:    '(D)
        Q = F:    H = N:        W = J:    GOSUB 13650   LUM4 = EPRN:    '(D1)
        ELL = LUM1 + LUM2 - LUM3 - LUM4
        RETURN

13770 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13550 AND GET FINAL RESULTS **************************
        'req: F, M, N, K, J (as in drawings), L (Luminance)

        IF N < 0 THEN N = 0
        Q = F:    H = M + N:    W = K:    GOSUB 13550:  LUM1 = ETILT:    '(AD)
        Q = F:    H = M + N:    W = J:    GOSUB 13550:  LUM2 = ETILT:    '(A1D1)
        Q = F:    H = N.        W = K:    GOSUB 13550:  LUM3 = ETILT:    '(D)
        Q = F:    H = N:        W = J:    GOSUB 13550:  LUM4 = ETILT:    '(D1)
        ELL = LUM1 + LUM2 - LUM3 - LUM4
        RETURN
```

```
13775 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13550 AND GET FINAL RESULTS ***********************
      'req: F, M, N, S, R (as in drawings)

      IF N < 0 THEN N = 0
      Q = F:      H = M + N:    W = S + R:     GOSUB 13550:  LUM1 = ETILT:    '(ABCD  look Murdock Ex2.13)
      Q = F:      H = M + N:    W = S:         GOSUB 13550:  LUM2 = ETILT:    '(BC)
      Q = F:      H = N:        W = S + R:     GOSUB 13550:  LUM3 = ETILT:    '(CD)
      Q = F:      H = N:        W = S:         GOSUB 13550:  LUM4 = ETILT:    '(C)
      ELL = LUM1 - LUM2 - LUM3 + LUM4
      RETURN

13780 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13550 AND GET FINAL RESULTS ***********************
      'req: EE, U1 , M, N, K, J (as in drawings), L (Luminance)

      IF N < 0 THEN N = 0
      Q = U1:     H = EE - N:   W = K:         GOSUB 13550:  LUM1 = ETILT:              '(AD)
      Q = U1:     H = EE - N:   W = J:         GOSUB 13550:  LUM2 = ETILT:              '(A1D1)
      Q = U1:     H = EE - N - M: W = K:       GOSUB 13550:  LUM3 = ETILT:    '(D)
      Q = U1:     H = EE - N - M: W = J:       GOSUB 13550:  LUM4 = ETILT.    '(D1)
      ELL = LUM1 + LUM2 - LUM3 - LUM4
      RETURN

13785 ****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES 13550 AND GET FINAL RESULTS ***********************
      'req: EE, U1, M, N, S, R (as in drawings)

      IF N < 0 THEN N = 0
      Q = U1:     H = EE - N:   W = S + R:     GOSUB 13550:  LUM1 = ETILT:    '(ABCD  look Murdock Ex2.13)
      Q = U1:     H = EE - N:   W = S:         GOSUB 13550:  LUM2 = ETILT:    '(BC)
      Q = U1:     H = EE - N - M: W = S + R:   GOSUB 13550:  LUM3 = ETILT:    '(CD)
      Q = U1:     H = EE - N - M: W = S:       GOSUB 13550:  LUM4 = ETILT:    '(C)
      ELL = LUM1 - LUM2 - LUM3 + LUM4
      RETURN

14300 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON OPPOSITE WALL (1) FROM VERTICAL WINDOW ***********************
      'req:

      F = WID:    M = WIDA(I, I2, 5):   N = WIDA(I, I2, 3):   Z = LENGTH / 2:   U = HIGHT / 2:   L = LWIND
      X1 = LENGTH - X - WIDA(I, I2, 4)
      X2 = X + WIDA(I, I2, 4)
EER1: IF Z < X OR Z > X2 THEN GOTO ER20: ELSE GOTO ER21:
ER20: R = WIDA(I, I2, 4)
      IF X > X1 THEN S = Z - R - X1 ELSE S = Z - R - X
      IF U > N AND U < M + N THEN GOSUB 13720: RETURN
      GOSUB 13700: RETURN
ER21: K = Z - X:   J = WIDA(I, I2, 4) - K:
      IF U > N AND U < M + N THEN GOSUB 13730: RETURN
      GOSUB 13710:   RETURN
14302 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON OPPOSITE WALL (2) FROM VERTICAL WINDOW ***********************
      Z = LENGTH / 6:      GOTO EER1:
14304 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON OPPOSITE WALL (3) FROM VERTICAL WINDOW ***********************
      Z = 5 * LENGTH / 6:      GOTO EER1:


14310 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE WALL (1) FROM VERTICAL WINDOW ***********************
      'req:

      X1 = LENGTH - X - WIDA(I, I2, 4):
      F = WID / 2:     M = WIDA(I, I2, 4):    N = X1:    Z = HIGHT / 2:    L = LWIND
EER2: IF Z < WIDA(I, I2, 3) OR Z < (HIGHT - WIDA(I, I2, 3) - WIDA(I, I2, 5)) THEN GOTO ER22: ELSE GOTO ER23:
ER22: R = WIDA(I, I2, 5)
      IF WIDA(I, I2, 3)>(HIGHT - WIDA(I, I2, 3) - WIDA(I, I2, 5)) THEN S = Z - R - (HIGHT - WIDA(I, I2, 3) - WIDA(I, I2, 5)) ELSE S=Z - R - WIDA(I, I2, 3)
      GOSUB 13750: RETURN
ER23: K = Z - WIDA(I, I2, 3):   J = WIDA(I, I2, 5) - K:
      GOSUB 13760:   RETURN
14312 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE WALL (2) FROM VERTICAL WINDOW ***********************
      F = WID / 6:       GOTO EER2:
14314 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE WALL (3) FROM VERTICAL WINDOW ***********************
      F = 5 * WID / 6:       GOTO EER2:


14320 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL (1) FROM VERTICAL WINDOW ***********************
      'req:

      X1 = LENGTH - X - WIDA(I, I2, 4)
      F = WID / 2:     M = WIDA(I, I2, 4):    N = X.    Z = HIGHT / 2:    L = LWIND
EER3: IF Z < WIDA(I, I2, 3) OR Z < (HIGHT - WIDA(I, I2, 3) - WIDA(I, I2, 5)) THEN GOTO ER22: ELSE GOTO ER23:
14322 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL (2) FROM VERTICAL WINDOW ***********************
      F = WID / 6:       GOTO EER3:
14324 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL (3) FROM VERTICAL WINDOW ***********************
      F = 5 * WID / 6:       GOTO EER3.
```

```
14330  ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON FLOOR (1) FROM VERTICAL WINDOW ******************************************
       'req:

       X1 = LENGTH - X - WIDA(I, I2, 4)
       F = WID / 2:      M = WIDA(I, I2, 5):      N = WIDA(I, I2, 3):      Z = LENGTH / 2:      L = LWIND
EER4:  IF Z < X OR Z < X1 THEN GOTO ER24:  ELSE GOTO ER25:
ER24:  R = WIDA(I, I2, 4)
       IF X1 > X THEN S = Z - R - X ELSE S = Z - R - X1
       GOSUB 13750: RETURN
ER25:  K = Z - X1:   J = WIDA(I, I2, 4) - K:      GOSUB 13760:   RETURN
14332  ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON FLOOR (2) FROM VERTICAL WINDOW ******************************************
       F = WID / 6:      GOTO EER4:
14334  ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON FLOOR (3) FROM VERTICAL WINDOW ******************************************
       F = 5 * WID / 6:      GOTO EER4:


14340  ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON CEILING (1) FROM VERTICAL WINDOW ****************************************
       'req:

       X1 = LENGTH - X - WIDA(I, I2, 4)
       F = WID / 2:      M = WIDA(I, I2, 5):      N = (HIGHT - WIDA(I, I2, 3) - WIDA(I, I2, 5)):      Z = LENGTH / 2:      L = LWIND
EER5:  IF Z < X OR Z < X1 THEN GOTO ER24:  ELSE GOTO ER25:
14342  ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON CEILING (2) FROM VERTICAL WINDOW ****************************************
       F = WID / 6:      GOTO EER5:
14344  ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON CEILING (3) FROM VERTICAL WINDOW ****************************************
       F = 5 * WID / 6:      GOTO EER5:

14350  ****************** SUBROUTINE TO CALCULATE ROOM SURFACES LUMINANCE DUE TO VERTICAL WINDOW ONLY ******************************

       GOSUB 14300: REFW(I, W3, SKCO) = ELL * RODA(I, W3, 4):              'OPPOSITE WALL (1)ILLUMINANCE
       GOSUB 14302: REFW(I, W3 + 6, SKCO) = ELL * RODA(I, W3, 4):          'OPPOSITE WALL (2)ILLUMINANCE
       GOSUB 14304: REFW(I, W3 + 12, SKCO) = ELL * RODA(I, W3, 4):         'OPPOSITE WALL (3)ILLUMINANCE

       GOSUB 14310: REFW(I, W2, SKCO) = ELL * RODA(I, W2, 4):              'RIGHT SIDE WALL (1) ILLUMINANCE
       GOSUB 14312: REFW(I, W2 + 6, SKCO) = ELL * RODA(I, W2, 4):          'RIGHT SIDE WALL (2) ILLUMINANCE
       GOSUB 14314: REFW(I, W2 + 12, SKCO) = ELL * RODA(I, W2, 4):         'RIGHT SIDE WALL (3) ILLUMINANCE

       GOSUB 14320: REFW(I, W4, SKCO) = ELL * RODA(I, W4, 4):              LEFT SIDE WALL (1)ILLUMINANCE
       GOSUB 14322: REFW(I, W4 + 6, SKCO) = ELL * RODA(I, W4, 4):          'LEFT SIDE WALL (2)ILLUMINANCE
       GOSUB 14324: REFW(I, W4 + 12, SKCO) = ELL * RODA(I, W4, 4):         'LEFT SIDE WALL (3)ILLUMINANCE

       GOSUB 14330. REFW(I, 5, SKCO) = ELL * RODA(I, 5, 4):                'FLOOR (1)ILLUMINANCE
       GOSUB 14332: REFW(I, 5 + 6, SKCO) = ELL * RODA(I, 5, 4):            'FLOOR (2)ILLUM'NANCE
       GOSUB 14334: REFW(I, 5 + 12, SKCO) = ELL * RODA(I, 5, 4):           'FLOOR (3)ILLUMINANCE

       GOSUB 14340: REFW(I,aw, SKCO) = ELL * RODA(I, 6, 4):                'CEILING (1)ILLUMINANCE
       GOSUB 14342: REFW(I, 6 + 6, SKCO) = ELL * RODA(I, 6, 4):            'CEILING (2)ILLUMINANCE
       GOSUB 14344· REFW(I, 6 + 12, SKCO) = ELL * RODA(I, 6, 4):           CEILING (3)ILLUMINANCE
       RETURN


14400  ****************** SUBROUTINE TO CALCULATE ROOM SURFACES ILLUMINANCE DUE TO TILT WINDOW ONLY ******************************
       'req:


14410  '------------------ FOR FLOOR ----------------------------------------------------------------------------------
       IF WALT = 0 THEN ETILT = 0: RETURN
       SLOP1 = PI / 2 - WALT
       X1 = LENGTH - X - WIDA(I, I2, 4)
       F = WID / 2:      M = WIDA(I, I2, 5):      N = WIDA(I, I2, 3):      Z = LENGTH / 2:      L = LWIND
EER6:  IF Z < X OR Z < X1 THEN GOTO ER26:  ELSE GOTO ER27:
ER26:  R = WIDA(I, I2, 4)
       IF X1 > X THEN S = Z - R - X ELSE S = Z - R - X1
       GOSUB 13775: RETURN
ER27:  K = Z - X1.   J = WIDA(I, I2, 4) - K:      GOSUB 13770:   RETURN
14412  '------------------ FOR FLOOR ----------------------------------------------------------------------------------
       F = WID / 6.      GOTO EER6:
14414  '------------------ FOR FLOOR ----------------------------------------------------------------------------------
       F = 5 * WID / 6:      GOTO EER6:


14420  '------------------ FOR CEILING --------------------------------------------------------------------------------
       IF WALT = 0 THEN ETILT = 0: RETURN
       SLOP1 = WALT + PI / 2:      L = LWIND:      Z = LENGTH / 2
       X1 = LENGTH - X - WIDA(I, I2, 4)
       F = WID / 2:      M = WIDA(I, I2, 5):      N = (HIGHT - WIDA(I, I2, 3) - WIDA(I, I2, 5)):      Z = LENGTH / 2·      L = LWIND
EER7:  IF Z < X OR Z < X1 THEN GOTO ER26:  ELSE GOTO ER27:
14422  '------------------ FOR CEILING --------------------------------------------------------------------------------
       F = WID / 6:      GOTO EER7:
```

```
14424  '------------------ FOR CEILING ------------------------------------------------------------------
       F = 5 * WID / 6:        GOTO EER7:


14430  '------------------ FOR OPPOSITE WALL -------------------------------------------------------------
       IF WALT = 0 OR WALT = PI / 2 THEN ETILT = 0: RETURN
       U = HIGHT / 2:   L = LWIND
       SLOP1 = WALT:   EE = WID / COS(SLOP1): U1 = WID * TAN(SLOP1) - U
       X1 = LENGTH - X - WIDA(I, I2, 4): X2 = X + WIDA(I, I2, 4)
       F = WID / 2:        M = WIDA(I, I2, 5):        N = WIDA(I, I2, 3):        Z = LENGTH / 2:        L = LWIND
EER8:  IF Z < X OR Z > X2 THEN GOTO ER28: ELSE GOTO ER285:
ER28:  R = WIDA(I, I2, 4)
       IF X1 > X THEN S = Z - R - X ELSE S = Z - R - X1
       GOSUB 13785: RETURN
ER285:  K = Z - X1:   J = WIDA(I, I2, 4) - K:        GOSUB 13780:    RETURN
14432  '------------------ FOR OPPOSITE WALL -------------------------------------------------------------
       Z = LENGTH / 6:        GOTO EER8:
14434  '------------------ FOR OPPOSITE WALL -------------------------------------------------------------
       Z = 5 * LENGTH / 6:                GOTO EER8:

14440  '-------------- SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE WALL FROM TILTED WINDOW --------------
       'req:

       X1 = LENGTH - X - WIDA(I, I2, 4)
       ZZ = Y * COS(WALT) - (HIGHT / 2):                    F = (WID / 2) - (ZZ + HIGHT / 2) * TAN(WALT)
       M = WIDA(I, I2, 4):     N = X1:     Z = HIGHT / 2:       L = LWIND
EER9:  IF Z < WIDA(I, I2, 3) OR Z < (HIGHT - WIDA(I, I2, 3) - WIDA(I, I2, 5)) THEN GOTO ER22: ELSE GOTO ER23:
14442  '-------------- SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE WALL FROM TILTED WINDOW --------------
       F = (WID / 6) - (ZZ + HIGHT / 2) * TAN(WALT):               GOTO EER9:
14444  '-------------- SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE WALL FROM TILTED WINDOW --------------
       F = (5 * WID / 6) - (ZZ + HIGHT / 2) * TAN(WALT):               GOTO EER9:


14450  '-------------- SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL FROM TILTED WINDOW --------------
       'req:

       X1 = LENGTH - X - WIDA(I, I2, 4)
       ZZ = Y * COS(WALT) - (HIGHT / 2):                    F = (WID / 2) - (ZZ + HIGHT / 2) * TAN(WALT)
       M = WIDA(I, I2, 4):     N = X:     Z = HIGHT / 2:       L = LWIND
EER10. IF Z < WIDA(I, I2, 3) OR Z < (HIGHT - WIDA(I, I2, 3) - WIDA(I, I2, 5)) THEN GOTO ER22: ELSE GOTO ER23:
14452  '-------------- SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL FROM TILTED WINDOW --------------
       F = (WID / 6) - (ZZ + HIGHT / 2) * TAN(WALT):               GOTO EER10:
14454  '-------------- SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL FROM TILTED WINDOW --------------
       F = (5 * WID / 6) - (ZZ + HIGHT / 2) * TAN(WALT):               GOTO EER10:


14500  '***************** SUBROUTINE TO CALCULATE ROOM SURFACES LUMINANCE DUE TO TILTED WINDOW ONLY *****************

       GOSUB 14430: REFW(I, W3, SKCO) = ELL * RODA(I, W3, 4):            'OPPOSITE WALL (1) ILLUMINANCE
       GOSUB 14432: REFW(I, W3 + 6, SKCO) = ELL * RODA(I, W3, 4):        'OPPOSITE WALL (2) ILLUMINANCE
       GOSUB 14434: REFW(I, W3 + 12, SKCO) = ELL * RODA(I, W3, 4):       'OPPOSITE WALL (3) ILLUMINANCE

       GOSUB 14440. REFW(I, W2, SKCO) = ELL * RODA(I, W2, 4):            'RIGHT SIDE WALL (1) ILLUMINANCE
       GOSUB 14442: REFW(I, W2 + 6, SKCO) = ELL * RODA(I, W2, 4):        'RIGHT SIDE WALL (2) ILLUMINANCE
       GOSUB 14444: REFW(I, W2 + 12, SKCO) = ELL * RODA(I, W2, 4):       'RIGHT SIDE WALL (3) ILLUMINANCE

       GOSUB 14450. REFW(I, W4, SKCO) = ELL * RODA(I, W4, 4):            'LEFT SIDE WALL (1) ILLUMINANCE
       GOSUB 14452: REFW(I, W4 + 6, SKCO) = ELL * RODA(I, W4, 4):        'LEFT SIDE WALL (2) ILLUMINANCE
       GOSUB 14454. REFW(I, W4 + 12, SKCO) = ELL * RODA(I, W4, 4):       'LEFT SIDE WALL (3) ILLUMINANCE

       GOSUB 14410. REFW(I, 5, SKCO) = ELL * RODA(I, 5, 4):              'FLOOR (1) ILLUMINANCE
       GOSUB 14412: REFW(I, 5 + 6, SKCO) = ELL * RODA(I, 5, 4):          'FLOOR (2) ILLUMINANCE
       GOSUB 14414: REFW(I, 5 + 12, SKCO) = ELL * RODA(I, 5, 4):         'FLOOR (3) ILLUMINANCE

       GOSUB 14420: REFW(I, 6, SKCO) = ELL * RODA(I, 6, 4):              'CEILING (1) ILLUMINANCE
       GOSUB 14422: REFW(I, 6 + 6, SKCO) = ELL * RODA(I, 6, 4):          'CEILING (2) ILLUMINANCE
       GOSUB 14424: REFW(I, 6 + 12, SKCO) = ELL * RODA(I, 6, 4):         'CEILING (3) ILLUMINANCE

       RETURN

14600  '***************** SUBROUTINE TO CALCULATE ROOM SURFACES INTERREFLECTIONS ***********************************

       EREF = 0:                  ROAVE1 = 0:          ATOT = 0
       FOR I5 = 1 TO 6:       ROAVE1 = ROAVE1 + RODA(I, I5, 4):   ATOT = ATOT + RODA(I, I5, 7)
       NEXT I5:        ROAVE = ROAVE1 / 6
       FOR I5 = 1 TO 6
       EREF1 = (REFS(I, I5, SKCO) + REFW(I, I5, SKCO)) * RODA(I, I5, 4) * RODA(I, I5, 7)
       EREF = EREF + EREF1
       NEXT I5
```

```
        EREF = EREF / ((1 - ROAVE) * ATOT)
        RETURN

14650 '***************** SUBROUTINE TO GET THE TOTAL LUMINANCE ON ROOM SURFACES ********************************

        FOR I5 = 1 TO 18
        IF I5 <= 6 THEN REFT(I, I5, SKCO) = (REFS(I, I5, SKCO) + REFW(I, I5, SKCO) + EREF) * RODA(I, I5, 4): GOTO RRR:
        IF 12 >= I5 AND I5 > 6 THEN REFT(I, I5, SKCO) = (REFS(I, I5, SKCO) + REFW(I, I5, SKCO) + EREF) * RODA(I, I5 - 6, 4): GOTO RRR:
        IF I5 > 12 THEN REFT(I, I5, SKCO) = (REFS(I, I5, SKCO) + REFW(I, I5, SKCO) + EREF) * RODA(I, I5 - 12, 4)
RRR:    NEXT I5:
        RETURN

14680 '***************** SUBROUTINE TO PREPARE REFERENCE POINTS VARIABLES ACCORDING TO WINDOW ********************
      '                       UNDER CONSIDERATION (W1, W2, W3, W4)
      '                           WHERE W1 IS THE WINDOW WALL

        IF WIDA(I, I2, 1) = 1 THEN REFE1 = REFP(I, I3, 1):           REFE2 = REFP(I, I3, 2)
        IF WIDA(I, I2, 1) = 2 THEN REFE1 = LENGTH - REFP(I, I3, 2):      REFE2 = REFP(I, I3, 1)
        IF WIDA(I, I2, 1) = 3 THEN REFE1 = LENGTH - REFP(I, I3, 1):      REFE2 = WID - REFP(I, I3, 2)
        IF WIDA(I, I2, 1) = 4 THEN REFE1 = REFP(I, I3, 2):           REFE2 = WID - REFP(I, I3, 1)
        RETURN

14700 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON REFERENCE POINT FROM VERTICAL WINDOW ***************
        'req: I, I3 LWIND, STT

        GOSUB 14710: EP1 = ELL: GOSUB 14712: EP12 = ELL: GOSUB 14714: EP14 = ELL:
        GOSUB 14720: EP2 = ELL: GOSUB 14722: EP22 = ELL: GOSUB 14724: EP24 = ELL:
        GOSUB 14730: EP3 = ELL: GOSUB 14732: EP32 = ELL: GOSUB 14734: EP34 = ELL:
        GOSUB 14740: EP4 = ELL: GOSUB 14742: EP42 = ELL: GOSUB 14744: EP44 = ELL:
        GOSUB 14750: EP5 = ELL: GOSUB 14760: EP6 = ELL:
        GOSUB 14770: EP7 = ELL: GOSUB 14772: EP72 = ELL: GOSUB 14774: EP74 = ELL:
        EEPP = EP1 + EP2 + EP3 + EP4 + EP5 - EP6 + EP7 + EP12 + EP22
        REFIL(I, I3, STT, SKCO) = EEPP + EP32 + EP42 + EP72 + EP14 + EP24 + EP34 + EP44 + EP74
        REFIL(I, I3, STT, SKCO) = INT(REFIL(I, I3, STT, SKCO) * 100) / 100
        RETURN

14710 '---------------- FROM CEILING (1) ---------------------------------------------------------------
        F = HIGHT - REFP(I, I3, 3):        M = WID / 3:     N = WID / 3: Z = LENGTH - REFE1
        U = WID - REFE2:   L = REFT(I, 6, SKCO):
ERR7:   J = LENGTH / 2:        K = J:
        IF U > N AND U < M + N THEN GOSUB 13730: RETURN
        GOSUB 13710:   RETURN

14712 '---------------- FROM CEILING (2) ---------------------------------------------------------------
        L = REFT(I, 6 + 6, SKCO):      N = 0
ERR9:   IF U > N AND U < M + N THEN GOSUB 13730: RETURN
        GOSUB 13710:   RETURN

14714 '---------------- FROM CEILING (3) ---------------------------------------------------------------
        L = REFT(I, 6 + 12, SKCO):     N = 2 * WID / 3
ERR14:  IF U > N AND U < M + N THEN GOSUB 13730: RETURN
        GOSUB 13710:   RETURN

14720 '---------------- FROM OPPOSITE WALL (1) ---------------------------------------------------------
        F = REFE2:         M = HIGHT - REFP(I, I3, 3):   L = REFT(I, W3, SKCO):    Z = REFE1:     ZZ1 = LENGTH / 3
ERR11:  IF 2 * ZZ1 > Z AND Z > ZZ1 THEN GOTO ERR1:   ELSE GOTO ERR2:
ERR1:   J = Z - ZZ1:   K = ZZ1 - J:   GOSUB 13760:   RETURN
ERR2:   R = ZZ1: IF Z > 2 * ZZ1 THEN S = Z - 2 * ZZ1 ELSE S = ZZ1 - Z:
        GOSUB 13750: RETURN

14722 '---------------- FROM OPPOSITE WALL (2) ---------------------------------------------------------
        L = REFT(I, W3 + 6, SKCO)
ERR12:  IF ZZ1 > Z THEN GOTO ERR3:  ELSE GOTO ERR4:
ERR3:   J = Z:   K = ZZ1 - J:   GOSUB 13760:   RETURN
ERR4:   R = ZZ1:  S = Z - R: GOSUB 13750: RETURN

14724 '---------------- FROM OPPOSITE WALL (3) ---------------------------------------------------------
        L = REFT(I, W3 + 12, SKCO)
ERR13:  IF 2 * ZZ1 < Z THEN GOTO ERR5:  ELSE GOTO ERR6:
ERR5:   J = Z - 2 * ZZ1: K = ZZ1 - J:   GOSUB 13760:   RETURN
ERR6:   R = ZZ1:  S = 2 * ZZ1 - Z: GOSUB 13750: RETURN


14730 '---------------- FROM RIGHT WALL (1) ------------------------------------------------------------
        F = LENGTH - REFE1:        M = HIGHT - REFP(I, I3, 3):    N = 0:
        L = REFT(I, W2, SKCO): Z = REFE2:      ZZ1 = WID / 3:  GOTO ERR11:
```

```
14732 '————————— FROM RIGHT WALL (2) —————————————————————
      L = REFT(I, W2 + 6, SKCO): GOTO ERR13:

14734 '————————— FROM RIGHT WALL (3) —————————————————————
      L = REFT(I, W2 + 12, SKCO): GOTO ERR12:

14740 '————————— FROM LEFT WALL (1) ——————————————————————
      F = REFE1:     M = HIGHT - REFP(I, I3, 3):   N = 0:
      L = REFT(I, W4, SKCO): Z = WID - REFE2:     ZZ1 = WID / 3: GOTO ERR11:

14742 '————————— FROM LEFT WALL (2) ——————————————————————
      L = REFT(I, W4 + 6, SKCO): GOTO ERR12:

14744 '————————— FROM LEFT WALL (3) ——————————————————————
      L = REFT(I, W4 + 12, SKCO): GOTO ERR13:


14750 '————————— FROM WINDOW —————————————————————————————
      X1 = LENGTH - X - WIDA(I, I2, 4)
      F = WID - REFE2:     M = WIDA(I, I2, 5):      N = WIDA(I, I2, 3) - REFP(I, I3, 3)
      Z = LENGTH - REFE1:   ZDD = REFE1
      L = LWIND
      IF ZDD < X OR Z < X1 THEN GOTO ER29: ELSE GOTO ER30:
ER29: R = WIDA(I, I2, 4)
      IF Z < X1 THEN S = X1 - Z ELSE S = X - ZDD
      GOSUB 13750:        RETURN
ER30: K = Z - X1:   J = WIDA(I, I2, 4) - K:    GOSUB 13760:        RETURN
      RETURN

14760 '————————— FROM WINDOW (ASSUMING WALL LUMINANCE) ———————————
      X1 = LENGTH - X - WIDA(I, I2, 4)
      F = WID - REFE2:     M = WIDA(I, I2, 5):      N = WIDA(I, I2, 3) - REFP(I, I3, 3)
      Z = LENGTH - REFE1:   ZDD = REFE1
      L = REFT(I, W1, SKCO)
      IF ZDD < X OR Z < X1 THEN GOTO ER31: ELSE GOTO ER32:
ER31: R = WIDA(I, I2, 4)
      IF Z < X1 THEN S = X1 - Z ELSE S = X - ZDD
      GOSUB 13750:      RETURN
ER32: K = Z - X1:   J = WIDA(I, I2, 4) - K:  GOSUB 13760:        RETURN
      RETURN

14770 '————————— FROM WINDOW WALL (1) —————————————————————
      F = WID - REFE2:      M = HIGHT - REFP(I, I3, 3):   N = 0:
      L = REFT(I, W1, SKCO):      Z = LENGTH - REFE1:    ZZ1 = LENGTH / 3: GOTO ERR11:

14772 '————————— FROM WINDOW WALL (2) —————————————————————
      L = REFT(I, W1 + 6, SKCO): GOTO ERR13:

14774 '————————— FROM WINDOW WALL (3) —————————————————————
      L = REFT(I, W1 + 12, SKCO): GOTO ERR12:


14800 ****************** SUBROUTINE TO PUT LAST SURFACES LUMINANCE VALUES TO ZERO ******************

      FOR I6 = 1 TO 6:   RODA(I, I6, 9) = 0:   RODA(I, I6, 10) = 0:   RODA(I, I6, 11) = 0
      NEXT I6:     RETURN


14900 ****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON REFERENCE POINT FROM TILTED WINDOW ******************
      'req: I, I3, LWIND, STT

      WIDM = WID - HIGHT / TAN(RODA(I, 1, 6))
      GOSUB 14910: GOSUB 14920: GOSUB 14930: GOSUB 14940: GOSUB 14950: GOSUB 14960: GOSUB 14970
      REFIL(I, I3, STT, SKCO) = EPT1 + EPT2 + EPT3 + EPT4 + EPT5 + EPT6 - EPT7
      REFIL(I, I3, STT, SKCO) = INT(REFIL(I, I3, STT, SKCO) * 100) / 100
      RETURN

14910 '————————— FROM CEILING —————————————————————————————
      F = HIGHT - REFE2:   M = LENGTH:    N = 0:       Z = REFE2
      U = REFE1:   L = REFT(I, 6, SKCO)
      IF Z < WIDM THEN GOTO ER33: ELSE GOTO ER34:
ER33: K = Z:  J = WIDM - K:  GOSUB 13730.   EPT1 = ELL:   RETURN
ER34: R = WIDM.     S = Z - R:     GOSUB 13720:  EPT1 = ELL:    RETURN

14920 '————————— FROM OPPOSITE WALL ——————————————————————
      F = REFE2:        M = HIGHT - REFP(I, I3, 3)·  N = 0:      J = REFE1:        K = LENGTH - J
      L = REFT(I, W3, SKCO)
      GOSUB 13760:        EPT2 = ELL
      RETURN
```

```
14930 '-------------- FROM RIGHT WALL ------------------------------------------
        F = LENGTH - REFE1:      M = HIGHT - REFP(I, I3, 3):   N = 0:   Z = REFE2
        L = REFT(I, W2, SKCO)
        IF Z > WIDM THEN GOTO ER35: ELSE GOTO ER36:
ER35:   R = WIDM:    S = Z - R:    GOSUB 13750:  EPT3 = ELL:   GOTO 14935
ER36:   K = Z:  J = WIDM - K:  GOSUB 13760:  EPT3 = ELL:   GOTO 14935

14935 '-------------- TRIANGLE PART OF THE RIGHT WALL --------------------------
        GOSUB 14730:  EPT3 = EPT3 + (EP3 - EPT3) / 2:     RETURN

14940 '-------------- FROM LEFT WALL -------------------------------------------
        F = REFE1:      M = HIGHT - REFP(I, I3, 3):   N = 0:   Z = REFE2
        L = REFT(I, W4, SKCO)
        IF Z < WIDM THEN GOTO ER37:  ELSE GOTO ER38:
ER37:   R = WIDM:    S = Z - R:    GOSUB 13750:  EPT4 = ELL:   GOTO 14945
ER38:   K = Z:  J = WIDM - K:  GOSUB 13760:  EPT4 = ELL:   GOTO 14945

14945 '-------------- TRIANGLE PART OF THE RIGHT WALL --------------------------
        GOSUB 14730:  EPT4 = EPT4 + (EP4 - EPT4) / 2:     RETURN

14950 '-------------- FROM WINDOW ----------------------------------------------
        SLOP1 = PI / 2 - WALT:
        X1 = LENGTH - X - WIDA(I, I2, 4)
        F = WID - REFE2:      M = WIDA(I, I2, 5):      N = WIDA(I, I2, 3) - REFP(I, I3, 3)
        Z = LENGTH - REFE1:   ZDD = REFE1
        L = LWIND
        IF ZDD < X OR Z < X1 THEN GOTO ER39:  ELSE GOTO ER40:
ER39:   R = WIDA(I, I2, 4)
        IF Z < X1 THEN S = X1 - Z ELSE S = X - ZDD
        GOSUB 13785:  EPT5 = ELL:      RETURN
ER40:   K = Z - X1:   J = WIDA(I, I2, 4) - K:    GOSUB 13780:      EPT5 = ELL:   RETURN
        RETURN

14960 '-------------- FROM WINDOW (ASSUMING WALL LUMINANCE) ---------------------
        L = REFT(I, W1, SKCO)
        SLOP1 = PI / 2 - WALT:
        X1 = LENGTH - X - WIDA(I, I2, 4)
        F = WID - REFE2:      M = WIDA(I, I2, 5):      N = WIDA(I, I2, 3) - REFP(I, I3, 3)
        Z = LENGTH - REFE1:   ZDD = REFE1
        IF ZDD < X OR Z < X1 THEN GOTO ER41:  ELSE GOTO ER42:
ER41:   R = WIDA(I, I2, 4)
        IF Z < X1 THEN S = X1 - Z ELSE S = X - ZDD
        GOSUB 13785:  EPT7 = ELL:      RETURN
ER42:   K = Z - X1:   J = WIDA(I, I2, 4) - K:    GOSUB 13780:      EPT7 = ELL:   RETURN
        RETURN

14970 '-------------- FROM TILTED WALL -----------------------------------------
        L = REFT(I, W1, SKCO):  SLOP1 = PI / 2 - WALT:
        F = WID - REFE2:     M = HIGHT - REFP(I, I3, 3):   N = 0:     J = REFE1:      K = LENGTH - J
        GOSUB 13780:        EPT6 = ELL
        RETURN


'*****************************************************************************
'********************* FILE READING AND WRITING ******************************
'*****************************************************************************

20100 '********** SUBROUTINE TO READ EXISTING FILE FROM DRIVE ********************

        OPEN FIL$ FOR INPUT AS #1
        INPUT #1, CITY.N$
        FOR I10 = 1 TO 7: FOR I11 = 1 TO 12: INPUT #1, BIDA(I10, I11): NEXT I11: NEXT I10
        NSP = BIDA(1, 1):      GOSUB 155
        FOR I10 = 1 TO NSP: FOR I11 = 1 TO 8:  FOR I12 = 1 TO 7:  INPUT #1, RODA(I10, I11, I12): NEXT I12: NEXT I11
             FOR I11 = 1 TO 10: FOR I12 = 1 TO 20: INPUT #1, WIDA(I10, I11, I12): NEXT I12: NEXT I11
             FOR I11 = 1 TO 10: FOR I12 = 1 TO 5:  INPUT #1, REFP(I10, I11, I12): NEXT I12: NEXT I11
        NEXT I10
         CLOSE
        OPEN CITY.N$ FOR INPUT AS #1
        FOR I10 = 1 TO 13: FOR I11 = 1 TO 7: INPUT #1, WEATH(I10, I11): NEXT I11: NEXT I10: CLOSE
        LAT = WEATH(13, 1): LONGT = WEATH(13, 2): STM = WEATH(13, 4)
        OPEN CITY.N$ + ".TEM" FOR INPUT AS #1
        FOR I10 = 1 TO 12: FOR I11 = 1 TO 72: INPUT #1, TEMP(I11, I10): NEXT I11: NEXT I10: CLOSE
        RETURN
```

```
****************************************************************************************************
**************************** ENERGY SAVING CALCULATION *********************************************
****************************************************************************************************
40005 ******************** SUBROUTINE TO CALCULATE THE JULIAN DAY **********************************
          'req: I,I3,RODA( ),DAY.D%, MON

          ON MON GOSUB 40010, 40020, 40030, 40040, 40050, 40060, 40070, 40080, 40090, 40100, 40110, 40120:   RETURN
40010     JULN = DAY.D%: RETURN
40020     JULN = DAY.D% + 31: RETURN
40030     JULN = DAY.D% + 59: RETURN
40040     JULN = DAY.D% + 90: RETURN
40050     JULN = DAY.D% + 120: RETURN
40060     JULN = DAY.D% + 151: RETURN
40070     JULN = DAY.D% + 181: RETURN
40080     JULN = DAY.D% + 212: RETURN
40090     JULN = DAY.D% + 243: RETURN
40100     JULN = DAY.D% + 273: RETURN
40110     JULN = DAY.D% + 304: RETURN
40120     JULN = DAY.D% + 334: RETURN

40135 ***** ********** REFERENCE BUILDING TOTAL ELECTRIC LIGHTING CONSUMPTION ****************************

          MLCONS = 0
          FOR I = 1 TO NSP:
          MLCONS = MLCONS + (RODA(I, 1, 1) * RODA(I, 2, 1) * BIDA(7, 6) / (RODA(1, 8, 1) * RODA(1, 8, 2) * RODA(1, 8, 3))): 'HOURLY LIGHT
                    CONSUMPTION (ELECTRICITY)
          '                          area           lux          COU          LLF          efficacy
          NEXT I
          HFLIGHT = MLCONS * (1 - (RODA(1, 8, 3) / 683)) * RODA(1, 8, 7): 'HEAT FROM LIGHT
          '                                    efficacy       plenum

40140 ' ******************** LIGHT CONSUMPTION WITH DAYLIGHT *********************************************
          FOR I = 1 TO NSP:
          FOR SKCO = 1 TO 3
               FOR STT = BIDA(7, 7) TO BIDA(7, 8)
                    FOR I3 = 1 TO RODA(I, 7, 5)
          IF REFIL(I, I3, STT, SKCO) * 1000 < BIDA(7, 6) THEN LCONS(STT, SKCO) = LCONS(STT, SKCO) + BIDA(7, 6) * REFP(I, I3, 5):    'LUMEN
          NEXT I3:
          NEXT STT:
          NEXT SKCO

          TMHR = 0:     TMSL = 0
          FOR STT = BIDA(7, 7) TO BIDA(7, 8):   EHLCONS = 0:
               FOR SKCO = 1 TO 3
                    EHLCONS = EHLCONS + LCONS(STT, SKCO) * WEATH(MON, SKCO) / (RODA(1, 8, 1) * RODA(1, 8, 2) * RODA(1, 8, 3)):    WATT
               NEXT SKCO
            EHFLIGHT = EHLCONS * (1 - (RODA(1, 8, 3) / 683)) * RODA(1, 8, 7): 'HEAT FROM LIGHT
            QG(STT) = HFLIGHT - EHFLIGHT
            TMHR = TMHR + QG(STT):     TMSL = TMSL + (MLCONS - EHLCONS)
          NEXT STT
          TMHR(I, MON) = TMHR * BIDA(2, MON):     'MONTHLY REDUCTION IN HEAT LOAD
          TMSL(I, MON) = TMSL * BIDA(2, MON):     'MONTHLY SAVING IN LIGHT
          NEXT I
          RETURN

30140 ********** REFERENCE BUILDING ENERGY REQUIREMENTS **********************************************

          RBHL1 = BIDA(7, 1) * WEATH(MON, 7) * 24:    'CONDUCTANCE
          RBHL2 = BIDA(7, 4) * .92 * WEATH(MON, 7) * 24:    'FLOOR
          RBHL3 = BIDA(7, 2) * .09 * 1.2 * WEATH(MON, 7) * 24:     'WALL SENSEBLE
          RBHL4 = BIDA(7, 2) * .09 * 2.608 * .5 * 24 * 30:  'WALL LATENT
          RBHL5 = BIDA(7, 3) * .5 * 1.2 * WEATH(MON, 7) * 24:     'WINDOWS SENSIBLE (INFULTRATION  = .5)
          RBHL6 = BIDA(7, 3) * .5 * 2.608 * .5 * 24 * 30: 'WINDOWS LATENT
          RBHLT(MON) = RBHL1 + RBHL2 + RBHL3 + RBHL4 + RBHL5 + RBHL6
          MLCONS = (BIDA(7, 8) - BIDA(7, 7)) * BIDA(2, MON) * BIDA(7, 6) * BIDA(7, 5) / (RODA(1, 8, 1) * RODA(1, 8, 2) * RODA(1, 8, 3)): 'MONTHLY LIGHT
                    CONSUMPTION (ELECTRICITY)
          '                 hour/day       day/month     lux     area      COU        LLF        efficacy

          HFLIGHT = MLCONS * (1 - (RODA(1, 8, 3) / 683)) * RODA(1, 8, 7): 'HEAT FROM LIGHT
          '                                    efficacy       plenum

          RBAUXIL(MON) = RBHLT(MON) - HFLIGHT:   'REFERENCE BUILDING AUXILIARY ENERGY
          RCOOLLOAD = 0
          IF HFLIGHT > RBHLT(MON) THEN RA92 = (HFLIGHT - RBHLT(MON)) / HFLIGHT: GOSUB 30170: RBAUXIL(MON) = RCOOLLOAD

          RBEREQ(MON) = RBAUXIL(MON) + L LCONS:    'REFERENCE BUILDING ENERGY REQUIREMENTS
```

```
30145   '************* THE ACTUAL BUILDING ENERGY REQUIREMENTS ****************************************

        BHLDT = 0: AR2 = 0: BHLD1 = 0
        FOR I = 1 TO NSP:    FOR I2 = 1 TO RODA(I, 7, 3): I9 = WIDA(I, I2, 1)
        AR1 = WIDA(I, I2, 4) * WIDA(I, I2, 5)
        BHLD1 = BHLD1 + (AR1 * WIDA(I, I2, 20)) * WEATH(MON, 7) * BIDA(7, 9):      'WINDOW CONDUCTANCE
        BHLD5 = BHLD5 + 2 * (WIDA(I, I2, 4) + WIDA(I, I2, 5)) * .5 * 1.2 * WEATH(MON, 7) * 24: 'WINDOWS SENSIBLE (INFULTRATION = 1)
        BHLD6 = BHLD6 + 2 * (WIDA(I, I2, 4) + WIDA(I, I2, 5)) * .5 * 2.808 * .5 * 24 * 30: 'WINDOWS LATENT
        AR2 = AR2 + AR1
        IF RODA(I, I9, 5) > 1 THEN GOTO REE1:

        BHLD2 = (RODA(I, I9, 1) * RODA(I, I9, 2) - AR2) * .5 * WEATH(MON, 7) * 24:      'WINDOW WALL CONDUCTANCE
        BHLD3 = (RODA(I, I9, 1) * RODA(I, I9, 2) - AR2) * .09 * 1.2 * WEATH(MON, 7) * 24: 'WALL SENSEBLE
        BHLD4 = (RODA(I, I9, 1) * RODA(I, I9, 2) - AR2) * .09 * 2.808 * .5 * 24 * 30.  'WALL LATENT
        BHLDT = BHLDT + BHLD2 + BHLD3 + BHLD4: AR2 = 0
REE1:   NEXT I2: NEXT I
        BHLDT(MON) = BHLDT + BHLD1 + BHLD5 + BHLD6 + RBHLT(MON)
        BTMSL(MON) = 0:     QS1(MON) = 0
        FOR I = 1 TO NSP: BTMSL(MON) = BTMSL(MON) + TMSL(I, MON): NEXT I
        BMLCONS(MON) = MLCONS - BTMSL(MON): 'BUILDING MONTHLY CONSUMPTION IN ELECTRIC LIGHTING
        IF BMLCONS(MON) < 0 THEN BMLCONS(MON) = 0

        BHFLIGHT(MON) = BMLCONS(MON) * (1 - (RODA(1, 8, 3) / 683)) * RODA(1, 8, 7):      'INTERNAL HEAT FROM LIGHT

        FOR I = 1 TO NSP:
        QS1(MON) = QS1(MON) + 30 * WASE(I, MON) * .278 * 10 ^ -3 * WEATH(MON, 6):     'SOLAR GAIN FROM WINDOW
        NEXT I

        BAUXIL(MON) = BHLDT(MON) - BHFLIGHT(MON) - QS1(MON):   'ACTUAL BUILDING AUXILIARY ENERGY
        COOLLOAD = 0
        IF BHFLIGHT(MON) + QS1(MON) > BHLDT(MON) THEN RA91 = (QS1(MON) + BHFLIGHT(MON) - BHLDT(MON)) / (QS1(MON) + BHFLIGHT(MON)):
        GOSUB 30155: BAUXIL(MON) = COOLLOAD

        BEREQ(MON) = BAUXIL(MON) + BMLCONS(MON):   'ACTUAL BUILDING ENERGY REQUIREMENTS

30160   '****************** SAVING DUE TO THE WINDOWS ***********************************************

        TENS(MON) = RBEREQ(MON) - BEREQ(MON):      'SAVING IN HEATING + LIGHTING
        RETURN

30155   '************************* COOLING LOAD FOR THE ACTUAL BUILDING ******************************

        AREAT = 0: FOR I = 1 TO NSP. AREAT = AREAT + RODA(I, 5, 7): NEXT I
        LCONS1 = BIDA(7, 6) * AREAT / (RODA(1, 8, 1) * RODA(1, 8, 2) * RODA(1, 8, 3)):
        HFL1 = LCONS1 * (1 - (RODA(1, 8, 3) / 683)) * RODA(1, 8, 7):

        FOR STT = 1 TO 24
        QG1(STT) = RA91 * (HFL1 - QG(STT))
        IF STT < BIDA(7, 7) OR STT > BIDA(7, 8) THEN QG1(STT) = 0
        NEXT STT

        FOR STT = 7 TO 72
        FOR I = 1 TO NSP: FOR I2 = 1 TO RODA(I, 7, 3)
        AR1 = WIDA(I, I2, 4) * WIDA(I, I2, 5)
        QW(STT) = AR1 * WIDA(I, I2, 20) * (TEMP(STT, MON) / 10 + WEATH(13, 5) - BIDA(4, MON)):  'WINDOW CONDUCTANCE
        AR2 = AR2 + AR1
        I4 = WIDA(I, I2, 1)
        IF RODA(I, I4, 5) > 1 THEN GOTO REE2:

        IF STT > 24 THEN GTT(I, I4, STT) = GTT(I, I4, STT - 24):
        GT(STT) = GTT(I, I4, STT) * WEATH(MON, 6) * .278 * 10 ^ -3:  'WALL
        TE(STT) = TEMP(STT, MON) / 10 + GT(STT) * .0395 + 4 * SIN(WIDA(I, I2, 6))
        SIGM1 = 0: SIGM2 = 0: SIGM3 = 0
        FOR III = 0 TO 6: SIGM1 = SIGM1 + BIDA(5, III + 1) * TE(STT - III): NEXT III
        FOR III = 1 TO 6: SIGM2 = SIGM2 + BIDA(6, III + 1) * QL(STT - III) / (RODA(I, 1, 1) * RODA(I, 1, 2) - AR2): NEXT III
        QL(STT) = (RODA(I, I4, 1) * RODA(I, I4, 2) - AR2) * (SIGM1 - SIGM2 - BIDA(4, MON) * BIDA(5, 8)): 'WALL OF THE WINDOW
        AR2 = 0
REE2:   NEXT I2: NEXT I: NEXT STT

        FOR STT = 1 TO 24: TRADT(STT) = 0: FOR I = 1 TO NSP: TRADT(STT) = TRADT(STT) + RADT(I, STT): NEXT I: NEXT STT

        FOR STT = 7 TO 72:
        IF STT > 24 THEN TRADT(STT) = TRADT(STT - 24)
        QS(STT) = RA91 * TRADT(STT) * WEATH(MON, 6) * .278 * 10 ^ -3      'SOLAR
        IF STT > 24 THEN QG1(STT) = QG1(STT - 24):       'LIGHTING
        QC(STT) = BIDA(1, 5) * QS(STT) + BIDA(1, 6) * QS(STT - 1) + BIDA(1, 7) * (QW(STT) + QL(STT)) + BIDA(1, 8) * (QW(STT - 1) + QL(STT - 1)) +
                  BIDA(1, 9) * QG1(STT) + BIDA(1, 10) * QG1(STT - 1) + BIDA(1, 11) * QG1(STT - 2) - BIDA(1, 12) * QC(STT _
                  - 1)
        NEXT STT: COOLLOAD = 0
```

```
FOR STT = 48 TO 72: COOLLOAD = COOLLOAD + QC(STT): NEXT STT
COOLLOAD = COOLLOAD / RODA(1, 8, 5) * 30
IF COOLLOAD < 0 THEN COOLLOAD = 0
RETURN

30170 '*********************** COOLING LOAD FOR THE REFERENCE BUILDING ****************************************

AREAT = 0: FOR I = 1 TO NSP: AREAT = AREAT + RODA(I, 5, 7): NEXT I
LCONS1 = BIDA(7, 6) * AREAT / (RODA(1, 8, 1) * RODA(1, 8, 2) * RODA(1, 8, 3)):
HFL1 = LCONS1 * (1 - (RODA(1, 8, 3) / 683)) * RODA(1, 8, 7):
FOR STT = 1 TO 24
QG1(STT) = RA92 * HFL1
IF STT < BIDA(7, 7) OR STT > BIDA(7, 8) THEN QG1(STT) = 0
NEXT STT

FOR STT = 7 TO 72:
IF STT > 24 THEN QG1(STT) = QG1(STT - 24):        'LIGHTING
RQC(STT) = BIDA(1, 9) * QG1(STT) + BIDA(1, 10) * QG1(STT - 1) + BIDA(1, 11) * QG1(STT - 2) - BIDA(1, 12) * RQC(STT - 1)
NEXT STT: RCOOLLOAD = 0
FOR STT = 48 TO 72: RCOOLLOAD = RCOOLLOAD + RQC(STT): NEXT STT
RCOOLLOAD = RCOOLLOAD / RODA(1, 8, 5) * BIDA(2, MON)
IF RCOOLLOAD < 0 THEN RCOOLLOAD = 0
RETURN

'*******************************************************************************************************
'********************* RUNNING THE PROGRAM ************************************************************
'*******************************************************************************************************

30100 '------- SAVING CALCULATIONS FOR A CERTAIN MONTH -----------------------------------------------
      'req:  MON

      OPEN "DATA.AAA" FOR INPUT AS #1:       INPUT #1, MON, FIL$: CLOSE : GOSUB 20100
30110 FILNM.SS$ = "SUN.SCR": CALL SCREENN: COLOR 4, 7: LOCATE 5, 45: PRINT MON
      DAY.D% = 21:   GOSUB 40005:   COUNT% = 0
      FOR STT = 1 TO 24
      GOSUB 10100:        GOSUB 10150:  IF SALT < 0 THEN GOTO RR4:
      GOSUB 10250:   COUNT% = COUNT% + 1
      OPEN "DATAB.AAA" FOR OUTPUT AS #1:       PRINT #1, SALT, SAZMS, JULN, STT, FIL$: CLOSE
      SHELL "BEAM"
      OPEN "RESB.AAA" FOR INPUT AS #1: FOR I = 1 TO NSP: FOR SKCO = 1 TO 2: INPUT #1, EDH(SKCO): FOR I5 = 1 TO 18: INPUT #1, REFS(I
          I5, SKCO): NEXT I5: NEXT SKCO: NEXT I: CLOSE
      FOR SKCO = 1 TO 3:     EDH1(STT, SKCO) = INT(EDH(SKCO) * 100) / 100: NEXT SKCO
      FOR I = 1 TO NSP:      FOR SKCO = 1 TO 3:     FOR I2 = 1 TO RODA(I, 7, 3):  GOSUB 13100: GOSUB 13200
      IF RODA(I, 7, 4) = 1 THEN GOTO RR1:  ELSE GOTO RR2:
RR1:  GOSUB 11000: GOSUB 11450: GOSUB 14500: GOSUB 14600: GOSUB 14650
      FOR I3 = 1 TO RODA(I, 7, 5): GOSUB 14680:   GOSUB 14900: NEXT I3:   GOTO RR3:
RR2:  GOSUB 11000: GOSUB 11450: GOSUB 14350: GOSUB 14600: GOSUB 14650
      FOR I3 = 1 TO RODA(I, 7, 5): GOSUB 14680:   GOSUB 14700: NEXT I3
RR3:  NEXT I2:       NEXT SKCO:        NEXT I
      Filenum = FREEFILE
      OPEN "DATAS.AAA" FOR OUTPUT AS #Filenum: PRINT #Filenum, SALT, SAZMS, JULN, MON, STT, FIL$: CLOSE
      SHELL "SOLAR":          'GOSUB 20690:
      OPEN "RESS.AAA" FOR INPUT AS #1: FOR I = 1 TO NSP: INPUT #1, RADT(I, STT): FOR I4 = 1 TO 4: INPUT #1, GTT(I, I4, STT): NEXT I4: NEXT
      I:        CLOSE
      FOR I = 1 TO NSP: WASE(I, MON) = WASE(I, MON) + RADT(I, STT)
      NEXT I
      COLOR 14, 7: LOCATE (11 + ABS(12 - STT)), (3 + 3 * STT): PRINT "*"
RR4:  COLOR 0, 7:    LOCATE 21, (2 + 3 * STT): PRINT ; STT
      NEXT STT: GOSUB 40135: GOSUB 30140        'GOSUB 20695
      IF ROUT.N% = 2 THEN RETURN
      OPEN "RESULT.AAA" FOR OUTPUT AS #1
      FOR STT = 1 TO 24: PRINT #1, STT, : FOR SKCO = 1 TO 3: PRINT #1, EKH1(STT, SKCO), EDH1(STT, SKCO), (EKH1(STT, SKCO) + EDH1(STT,
      SKCO)),      : NEXT SKCO: PRINT #1, : NEXT STT: PRINT #1,
      FOR STT = 1 TO 24:       FOR I = 1 TO NSP: PRINT #1, STT, RADT(I, STT), : NEXT I: PRINT #1, :       NEXT STT:   PRINT #1,
      FOR SKCO = 1 TO 3:  FOR I = 1 TO NSP: FOR STT = 1 TO 24: PRINT #1, STT, : FOR I3 = 1 TO RODA(I, 7, 5): PRINT #1, REFIL(I, I3, STT, SKCO),
      NEXT I3: PRINT #1, : NEXT STT: PRINT #1, : NEXT I:        NEXT SKCO:
      PRINT #1, NSP: CLOSE
      OPEN "RESULT1.AAA" FOR OUTPUT AS #1
      PRINT #1, MON, TENS(MON) / 1000, BTMSL(MON) / 1000, RBHLT(MON) / 1000, RBAUXIL(MON) / 1000, RBEREQ(MON) / 1000, HFLIGHT / 1000,
      MLCONS / 1000, BHLDT(MON) / 1000, BAUXIL(MON) / 1000, BEREQ(MON) / 1000, BHFLIGHT(MON) / 1000, BMLCONS(MON) / _
      1000, QS1(MON) / 1000  CLOSE
      END

30150 '------------------------------------ CALCULATE THE SAVING FOR A WHOLE YEAR ----------------------------

      OPEN "DATA.AAA" FOR INPUT AS #15:      INPUT #15, FIL$: CLOSE : GOSUB 20100
      T.SAV = 0: RBAUXILT = 0: BTMSLT = 0: RBHLTT = 0
      FOR MON = 1 TO 12: GOSUB 30110:
      T.SAV = T.SAV + TENS(MON) / 1000000
```

```
BTMSLT = BTMSLT + BTMSL(MON) / 1000000
RBHLTT = RBHLTT + RBHLT(MON) / 1000000
RBAUXILT = RBAUXILT + RBAUXIL(MON) / 1000000
RBEREQT = RBEREQT + RBEREQ(MON) / 1000000
HFLIGHTT = HFLIGHTT + HFLIGHT / 1000000
MLCONST = MLCONST + MLCONS / 1000000
BHLDTT = BHLDTT + BHLDT(MON) / 1000000
BAUXILT = BAUXILT + BAUXIL(MON) / 1000000
BEREQT = BEREQT + BEREQ(MON) / 1000000
BHFLIGHTT = BHFLIGHTT + BHFLIGHT(MON) / 1000000
BMLCONST = BMLCONST + BMLCONS(MON) / 1000000
QS1T = QS1T + QS1(MON) / 1000000
: NEXT MON

OPEN "RESULT.AAA" FOR OUTPUT AS #15: PRINT #15, T.SAV, BTMSLT, RBHLTT, RBAUXILT, RBEREQT, HFLIGHTT, MLCONST, BHLDTT,
BAUXILT, BEREQT, BHFLIGHTT, BMLCONST, QS1T: CLOSE

OPEN "RESULT1.AAA" FOR OUTPUT AS #15:
PRINT #15, 2, "" REFERENCE BUILDING ""
FOR MON = 1 TO 12: PRINT #15, MON, RBHLT(MON) / 1000, HFLIGHT / 1000
NEXT MON: PRINT #15, "HEAT LOSS FROM THE ENVELOPE": PRINT #15, "INTERNAL HEAT GAIN FROM LIGHT": CLOSE

OPEN "RESULT2.AAA" FOR OUTPUT AS #15:
PRINT #15, 3, "" REFERENCE BUILDING ""
FOR MON = 1 TO 12: PRINT #15, MON, RBAUXIL(MON) / 1000, RBEREQ(MON) / 1000, MLCONS / 1000
NEXT MON: PRINT #15, "AUXILIARY ENERGY FOR HEAT.& COOL.": PRINT #15, "TOTAL ENERGY CONSUMED": PRINT #15, "ENERGY
CONSUMED FOR LIGHTING": CLOSE

OPEN "RESULT3.AAA" FOR OUTPUT AS #15:
PRINT #15, 4, "" ACTUAL BUILDING ""
FOR MON = 1 TO 12: PRINT #15, MON, BHLDT(MON) / 1000, BHFLIGHT(MON) / 1000, QS1(MON) / 1000, QS1(MON) / 1000 + BHFLIGHT(MON)
/ 1000
NEXT MON: PRINT #15, "HEAT LOSS FROM THE ENVELOPE": PRINT #15, "INTERNAL HEAT GAIN FROM LIGHT": PRINT #15, "SOLAR ENERGY
HEATING THE BUILDING": PRINT #15, "TOTAL HEAT GAIN IN THE BUILDING": CLOSE

OPEN "RESULT4.AAA" FOR OUTPUT AS #15:
PRINT #15, 3, "" ACTUAL BUILDING ""
FOR MON = 1 TO 12: PRINT #15, MON, BAUXIL(MON) / 1000, BEREQ(MON) / 1000, BMLCONS(MON) / 1000
NEXT MON: PRINT #15, "AUXILIARY ENERGY FOR HEAT.& COOL.": PRINT #15, "TOTAL ENERGY CONSUMED": PRINT #15, "ENERGY
CONSUMED FOR LIGHTING": CLOSE

OPEN "RESULT5.AAA" FOR OUTPUT AS #15:
PRINT #15, 4, "" ANNUAL SAVING ""
FOR MON = 1 TO 12: PRINT #15, MON, RBEREQ(MON) / 1000, BEREQ(MON) / 1000, TENS(MON) / 1000, "0"
NEXT MON: PRINT #15, "R.BUILDING ENERGY CONSUMPTION": PRINT #15, "ACTUAL BUILDING ENERGY CONSUMPTION": PRINT #15,
"SAVING IN ENERGY": PRINT #15, "ZERO LINE": CLOSE
END

30200 '------ SKY ILLUMINANCE ON HORIZONTAL SURFACE ------------------------------------------------
      OPEN "DATA.AAA" FOR INPUT AS #20.     INPUT #20, JULN, ST, LAT, LONGT, SKCO: CLOSE
      LAT = LAT * PI / 180:   LONGT = LONGT * PI / 180
      I4.M% = 1: GOSUB 10150: IF SALT < 0 THEN EKH = 0: GOTO 30205
      GOSUB 11000
30205 OPEN "RESULT.AAA" FOR OUTPUT AS #20:     PRINT #20, EKH:  CLOSE : END

30250 '------ SKY ILLUMINANCE ON VERTICAL SURFACE ------------------------------------------------
      OPEN "DATA.AAA" FOR INPUT AS #25:     INPUT #25, JULN, ST, LAT, LONGT, SKCO, AZM: CLOSE
      LAT = LAT * PI / 180:   LONGT = LONGT * PI / 180:     AZM = AZM * PI / 180:   SLOP = PI / 2
      I4.M% = 1: GOSUB 10150: IF SALT < 0 THEN LWIND = 0: GOTO 30255
      GOSUB 11450
30255 OPEN "RESULT.AAA" FOR OUTPUT AS #25:     PRINT #25, LWIND:    CLOSE : END

30300 '------ SKY ILLUMINANCE ON TILTED SURFACE ------------------------------------------------
      OPEN "DATA.AAA" FOR INPUT AS #30:     INPUT #30, JULN, ST, LAT, LONGT, SKCO, AZM: CLOSE
      LAT = LAT * PI / 180:   LONGT = LONGT * PI / 180:     AZM = AZM * PI / 180:   SLOP = PI / 2 - (SLOP * PI / 180)
      I4.M% = 1: GOSUB 10150: IF SALT < 0 THEN LWIND = 0: GOTO 30305
      GOSUB 11450
30305 OPEN "RESULT.AAA" FOR OUTPUT AS #30:     PRINT #30, LWIND   CLOSE : END

30350 '------ HOUR ANGLE AND SUNSET HOUR ANGLE ------------------------------------------------
      OPEN "DATA.AAA" FOR INPUT AS #35:     INPUT #35, JULN, ST, LAT, LONGT:  CLOSE
      LAT = LAT * PI / 180:   LONGT = LONGT * PI / 180
      I4.M% = 1: GOSUB 10150: GOSUB 10250
      HANG = HANG * 180 / PI:        SSHANG = SSHANG * 180 / PI
      OPEN "RESULT.AAA" FOR OUTPUT AS #35:     PRINT #35, HANG, SSHANG: CLOSE : END

30400 '------ SOLAR ILLUMINATION ON A SURFACE ------------------------------------------------
      OPEN "DATA.AAA" FOR INPUT AS #40:     INPUT #40, JULN, ST, LAT, LONGT, SKCO, AZM, SLOP: CLOSE
      LAT = LAT * PI / 180:   LONGT = LONGT * PI / 180:     AZM = AZM * PI / 180:   SLOP = PI / 2 - (SLOP * PI / 180)
      I4.M% = 1: GOSUB 10150:       IF SALT < 0 THEN EDT = 0: GOTO 30405
```

```
        AZMD = ABS(SAZMS - AZM)
        GOSUB 5200:    GOSUB 13400
        EDT = EDH * COS(SLOP) + EDV * SIN(SLOP): IF EDT < .001 THEN EDT = 0
30405   OPEN "RESULT.AAA" FOR OUTPUT AS #40:    PRINT #40, EDT:  CLOSE : END

30450 '———— INCIDENT ANGLE ON A SURFACE ————————————————————————
        OPEN "DATA.AAA" FOR INPUT AS #45:       INPUT #45, JULN, ST, LAT, LONGT, AZM, SLOP: CLOSE
        LAT = LAT * PI / 180:  LONGT = LONGT * PI / 180:     AZM = AZM * PI / 180:  SLOP = PI / 2 - (SLOP * PI / 180)
        I4.M% = 1: GOSUB 10150:       IF SALT < 0 THEN INANG = 0: GOTO 30455
        AZMD = ABS(SAZMS - AZM):        GOSUB 5200
        INANG = INANG * 180 / PI
30455   OPEN "RESULT.AAA" FOR OUTPUT AS #45:    PRINT #45, INANG: CLOSE : END

30500 '———— POSITION OF THE SUN ————————————————————————————
        OPEN "DATA.AAA" FOR INPUT AS #50:       INPUT #50, JULN, ST, LAT, LONGT:  CLOSE
        LAT = LAT * PI / 180:  LONGT = LONGT * PI / 180
        I4.M% = 1: GOSUB 10150: IF SALT < 0 THEN SALT = 0: SAZMS = 0: GOTO 30505
        SALT = SALT * 180 / PI:        SAZMS = SAZMS * 180 / PI
30505   OPEN "RESULT.AAA" FOR OUTPUT AS #50:    PRINT #50, SALT, SAZMS: CLOSE : END


        SUB SCREENN
        DEF SEG = &HB800: BLOAD FILNM.SS$, 0: DEF SEG ' Load screen picture
        END SUB
```

Direct Beam Routine

```
'******************************************************************************************************
'****************************** PROGRAM TO CALCULATE THE EFFECT OF SOLAR SPOT IN A BUILDING ************
'******************************************************************************************************
DECLARE SUB OVERHANG (SALT, SAZMS, ARR, FI)

        DIM SHARED BIDA(7, 12), SAZMS, SALT, FI, I2, I
        OPEN "DATAB.AAA" FOR INPUT AS #1:     INPUT #1, SALT, SAZMS, JULN, STT, FIL$: CLOSE
        PI = 3.141592653#:     I4, M% = 0.     GOSUB 20100:    GOTO 158
155     DIM SHARED RODA(NSP, 8, 16), WIDA(NSP, 10, 20), REFP(NSP, 10, 6), REFS(NSP, 18, 3), EDH(3): RETURN
158     FOR I = 1 TO NSP:     FOR SKCO = 1 TO 2:     FOR I2 = 1 TO RODA(I, 7, 3)
        GOSUB 10300: GOSUB 13100: GOSUB 13200: GOSUB 13450: GOSUB 13300: GOSUB 13500
        NEXT I2:     NEXT SKCO:     NEXT I
        OPEN "RESB.AAA" FOR OUTPUT AS #1: FOR I = 1 TO NSP: FOR SKCO = 1 TO 2: PRINT #1, EDH(SKCO); : FOR I5 = 1 TO 18: PRINT #1, REFS(I,
        I5, SKCO); : NEXT I5: PRINT #1, : NEXT SKCO: NEXT I
        END

10300 '********* SUBROUTINE TO CALCULATE SUN POSITION RELATIVE TO A WINDOW  (SPOS) ***************************
      '  req: SAZMS, I & I2 (for window matrix)

        IF SAZMS <= 0 AND WIDA(I, I2, 19) <= 0 THEN GOTO DE1:
        IF SAZMS <= 0 AND WIDA(I, I2, 19) > 0 THEN GOTO DE2:
        IF SAZMS > 0 AND WIDA(I, I2, 19) <= 0 THEN GOTO DE3:
        IF SAZMS > 0 AND WIDA(I, I2, 19) > 0 THEN GOTO DE1:
DE1:    IF (WIDA(I, I2, 19) - SAZMS) < 0 THEN SPOS = 2 ELSE SPOS = 1
        GOTO DE4:
DE2:    IF SAZMS <= -PI / 2 THEN SPOS = 2 ELSE SPOS = 1: GOTO DE4:
DE3:    IF SAZMS >= PI / 2 THEN SPOS = 1 ELSE SPOS = 2
DE4:    RETURN
'******************************************************************************************************

5200  '******** SUBROUTINE TO CALCULATE INCIDENT ANGLE ON A SURFACE ******************************************
      '   (req:- SLOP,  SALT)

        RA9 = COS(SALT) * COS(AZMD) * SIN(SLOP) + SIN(SALT) * COS(SLOP)
        INANG = ATN(SQR(1 - RA9 ^ 2) / RA9)
        IF INANG < 0 THEN INANG = INANG + PI
        RETURN

'******************************************************************************************************
'******************************* DIRECT BEAM CALCULATIONS *********************************************
'******************************************************************************************************

13100 '****************** SUBROUTINE TO PREPARE WALLS # ACCORDING TO WINDOW UNDER CONSIDERATION (W1, W2, W3, W4) ********************
      '                      WHERE W1 IS THE WINDOW WALL
      '  req:

        IF WIDA(I, I2, 1) = 1 THEN W1 = 1:    W2 = 2:     W3 = 3:     W4 = 4
        IF WIDA(I, I2, 1) = 2 THEN W1 = 2:    W2 = 3:     W3 = 4:     W4 = 1
        IF WIDA(I, I2, 1) = 3 THEN W1 = 3:    W2 = 4:     W3 = 1:     W4 = 2
        IF WIDA(I, I2, 1) = 4 THEN W1 = 4:    W2 = 1:     W3 = 2:     W4 = 3
        RETURN

13200 '****************** SUBROUTINT TO PREPARE WINDOW VARIABLES FOR SUBROUTINE 13300 *************************
      '  req: I, I2, W1, W2, W3, W4 (WALLS #)

        WALT = WIDA(I, I2, 6):        Y = WIDA(I, I2, 3):    F = WIDA(I, I2, 5)
        M = WIDA(I, I2, 4):           X = WIDA(I, I2, 2):    WID = RODA(I, W2, 1)
        LENGTH = RODA(I, W1, 1):      HIGHT = RODA(I, W2, 2)
        RETURN

13300 '****************** SUBROUTINE TO CALCULATE PARAMETERS OF SUN SPOT CALCULATIONS IN A ROOM FROM A WINDOW ********************
      '  req: WALT, SALT, Y, F, M, X, WID=WIDTH, LENGTH, HIGHT, I & I2 (for WIDA), SPOS

        IF SPOS = 1 THEN X1 = LENGTH - M - X ELSE X1 = X
        AZMD = ABS(SAZMS - WIDA(I, I2, 19))
        IF AZMD > PI THEN AZMD = (2 * PI) - AZMD
        IF COS(AZMD) <= .0001 THEN RETURN:     '(no sun enter through this window)
        CITA = ATN(TAN(SALT) / COS(AZMD))
        S = Y * SIN(WALT) + Y * COS(WALT) / TAN(CITA)
        S1 = (Y + F) * (SIN(WALT) + COS(WALT) / TAN(CITA))
        Z = Y * COS(WALT) * TAN(AZMD) / TAN(CITA)
        ZD = (Y + F) * COS(WALT) * TAN(AZMD) / TAN(CITA)
        R = ZD - Z
        A = Y * SIN(WALT)
```

```
        AD = (Y + F) * SIN(WALT)
        B = Y * COS(WALT) / TAN(CITA)
        BD = (Y + F) * COS(WALT) / TAN(CITA)
        IF AZMD = 0 THEN GOTO EE1:
        P1 = AD + X1 / TAN(AZMD)
        P2 = A + X1 / TAN(AZMD)
        P3 = A + (M + X1) / TAN(AZMD)
        P4 = AD + (M + X1) / TAN(AZMD)
EE1:    Z1 = (AD + BD - P1) * TAN(CITA)
        Z2 = (A + B - P2) * TAN(CITA)
        Z3 = (A + B - P3) * TAN(CITA)
        Z4 = (AD + BD - P4) * TAN(CITA)
        RETURN

13500 '*********************** SUBROUTINE TO DETERMINE WHICH CASE SHOULD BE APPLIED ***********************************************************
      '  req: Z, X1, ZD, P1, P2, P3, P4, S, S1, WID=WIDTH, SPOS (1=east 2=west)

        AZMD = ABS(SAZMS - WIDA(I, I2, 19))
        IF AZMD > PI THEN AZMD = (2 * PI) - AZMD
        IF COS(AZMD) <= .0001 THEN RETURN:     '(no sun enter through this window)
        IF Z >= M + X1 THEN GOTO E1:
        IF S >= WID THEN GOTO E1:
        IF P2 <= WID THEN GOTO E3:
        IF S1 >= WID THEN GOTO EC4: ELSE GOTO EC1:
E1:     IF P2 >= WID THEN GOTO EC2:
        IF P4 >= WID THEN GOTO E2: ELSE GOTO EC3:
E2:     IF P3 >= WID THEN GOTO E6:
        IF P1 >= WID THEN GOTO EC13: ELSE GOTO EC14:
E3:     IF S1 >= WID THEN GOTO E4:
        IF ZD >= M + X1 THEN GOTO EC5:
        IF Z >= X1 THEN GOTO EC6: ELSE GOTO EC7:
E4:     IF Z >= X1 THEN GOTO E5:
        IF P1 >= WID THEN GOTO EC9: ELSE GOTO EC8:
E5:     IF P1 >= WID THEN GOTO EC11: ELSE GOTO EC10:
E6:     IF P1 >= WID THEN GOTO EC15: ELSE GOTO EC12:

EC1:  '-------------- SUBROUTINE FOR CASE # 1 --------------------------------------------------------------------------------

        FX1 = LENGTH - X1 - M + Z + R / 2:     FX2 = M:      FX3 = LENGTH - FX1 - FX2
        FY3 = S:        FY2 = S1 - S:  FY1 = WID - FY3 - FY2
        SZ1 = 0:        SZ2 = 0:       SZ3 = HIGHT - SZ1 - SZ2
        SY3 = 0:        SY2 = 0:       SY1 = WID - SY2 - SY3
        OX1 = 0:        OX2 = 0:       OX3 = LENGTH - OX1 - OX2
        OZ1 = 0:        OZ2 = 0:       OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC2:  '-------------- SUBROUTINE FOR CASE # 2 --------------------------------------------------------------------------------

        IF AZMD = 0 THEN RD = 0 ELSE RD = (P1 - P2) * TAN(AZMD)
        FX1 = 0:        FX2 = 0:       FX3 = LENGTH - FX1 - FX2
        FY3 = S:        FY2 = 0:       FY1 = WID - FY3 - FY2
        SZ1 = 0:        SZ2 = 0:       SZ3 = HIGHT - SZ1 - SZ2
        SY3 = 0:        SY2 = 0:       SY1 = WID - SY2 - SY3
        IF SPOS = 1 THEN OX1 = LENGTH - RD / 2 - M - (P2 - WID) * TAN(AZMD) ELSE OX1 = RD / 2 + (P2 - WID) * TAN(AZMD)
            OX2 = M:        OX3 = LENGTH - OX1 - OX2
        OZ1 = (S - WID) * TAN(CITA):    OZ2 = (S1 - S) * TAN(CITA):    OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC3:  '-------------- SUBROUTINE FOR CASE # 3 --------------------------------------------------------------------------------

        FX1 = 0:        FX2 = 0:       FX3 = LENGTH - FX1 - FX2
        FY3 = 0.        FY2 = 0:       FY1 = WID - FY3 - FY2
        SZ1 = (Z1 + Z3 - F) / 2:       SZ2 = F:             SZ3 = HIGHT - SZ1 - SZ2
        SY3 = (P4 + P2) / 2 - (P3 - P2) / (2 * COS(CITA)):     SY2 = (P3 - P2) / COS(CITA):      SY1 = WID - SY2 - SY3
        OX1 = 0:        OX2 = 0.       OX3 = LENGTH - OX1 - OX2
        OZ1 = 0:        OZ2 = 0:       OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC4:  '-------------- SUBROUTINE FOR CASE # 4 --------------------------------------------------------------------------------

        R1 = R - ((S1 - WID) * R / (S1 - S))
        IF AZMD = 0 THEN MY1 = X1 ELSE MY1 = (P1 - WID) * TAN(AZMD)
        R2 = MY1 + R1 + Z - X1
        IF SPOS = 1 THEN FX1 = X + Z + R1 / 2 ELSE FX1 = X - Z - R1 / 2
        IF SPOS = 1 THEN OX1 = LENGTH - (MY1 + M) + R2 / 2 ELSE OX1 = MY1 - R2 / 2
            FX2 = M:        FX3 = LENGTH - FX1 - FX2
        FY3 = S:        FY2 = WID - S:  FY1 = WID - FY3 - FY2
        SZ1 = 0:        SZ2 = 0:       SZ3 = HIGHT - SZ1 - SZ2
        SY3 = 0:        SY2 = 0.       SY1 = WID - SY2 - SY3
                OX2 = M:        OX3 = LENGTH - OX1 - OX2
```

```
        OZ1 = 0:      OZ2 = (S1 - WID) * TAN(CITA):   OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC5:  '------------ SUBROUTINE FOR CASE # 5 ------------------------------------------

        R1 = M + X1 - Z
        AB = (S1 - S) / (1 + (R - R1) / R1)
        IF WALT = 0 THEN HHH = Z4 ELSE HHH = (P4 - S - AB) / SIN(WALT)
        HA = AB * COS(WALT)
        AREAS = (S - P2) * F * COS(WALT - CITA) / COS(CITA) + (F + HHH) * HA / 2
        H = (3 * Z1 + Z2 + Z4) / 5:    PAV = (P1 + P2 + P4) / 3
        MZZ = AREAS / H
        IF SPOS = 1 THEN FX1 = LENGTH - R1 / 2 ELSE FX1 = 0
            FX2 = R1 / 2:  FX3 = LENGTH - FX1 - FX2
        FY3 = S:       FY2 = AB:      FY1 = WID - FY3 - FY2
        SZ1 = Z2 / 2:  SZ2 = H:       SZ3 = HIGHT - SZ1 - SZ2
        SY3 = PAV - MZZ / 2:          SY2 = MZZ:      SY1 = WID - SY2 - SY3
        OX1 = 0:       OX2 = 0:       OX3 = LENGTH - OX1 - OX2
        OZ1 = 0:       OZ2 = 0:       OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC6:  '------------ SUBROUTINE FOR CASE # 6 ------------------------------------------

        IF SPOS = 1 THEN FX1 = LENGTH - X1 - M + Z + R / 2 ELSE FX1 = 0
        IF SPOS = 1 THEN FX2 = LENGTH - FX1 ELSE FX2 = X1 + M - Z - R2
                         FX3 = LENGTH - FX1 - FX2
        FY3 = S:       FY2 = S1 - S:  FY1 = WID - FY3 - FY2
        SZ1 = 0:       SZ2 = (Z1 + Z2) / 2:  SZ3 = HIGHT - SZ1 - SZ2
        SY3 = P2:      SY2 = (S1 + S - P1 - P2) * (S1 - S) * TAN(CITA) / (Z1 + Z2):    SY1 = WID - SY2 - SY3
        OX1 = 0:       OX2 = 0:       OX3 = LENGTH - OX1 - OX2
        OZ1 = 0:       OZ2 = 0:       OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC7:  '------------ SUBROUTINE FOR CASE # 7 ------------------------------------------

        PP = S + ((X1 - Z) * (S1 - S) * TAN(CITA) / (F * COS(WALT) * TAN(AZMD)))
        AREAS = M * (PP - S) + ((M + (P4 - S1) * TAN(AZMD)) / 2) * (S1 - PP)
        MZ = 3 * (S1 - PP) / 4
        IF SPOS = 1 THEN FX1 = X + Z + R / 2 ELSE FX2 = X - Z - R / 2
                    FX2 = AREAS / (S1 - S):         FX3 = LENGTH - FX1 - FX2
        FY3 = S:       FY2 = S1 - S:        FY1 = WID - FY3 - FY2
        SZ1 = 0:       SZ2 = 2 * Z1 / 3:    SZ3 = HIGHT - SZ1 - SZ2
        SY3 = (PP + S1 - MZ) / 2:           SY2 = MZ:     SY1 = WID - SY2 - SY3
        OX1 = 0:       OX2 = 0:             OX3 = LENGTH - OX1 - OX2
        OZ1 = 0:       OZ2 = 0:             OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC8:  '------------ SUBROUTINE FOR CASE # 8 ------------------------------------------

        PP = S + ((X1 - Z) * (S1 - S) * TAN(CITA) / (F * COS(WALT) * TAN(AZMD)))
        AA = R * (WID - PP) / (S1 - S)
        AREAS = M * (PP - S) + (2 * M - AA) * (WID - PP) / 2
        R4 = AREAS / (WID - S)
        H1 = 2 * Z1 / 3:        H = (S1 - WID) * TAN(CITA)
        AREAS1 = .5 * (S1 - PP) * Z1 - .5 * (S1 - WID) * H
        R1 = R - ((S1 - WID) * R / (S1 - S))
        AB = (P4 - WID) * TAN(AZMD):          AA1 = X1 + M - Z - R1
        IF SPOS = 1 THEN FX1 = LENGTH - R4 - (X1 - Z) / 2 ELSE FX1 = (X1 - Z) / 2
        IF SPOS = 1 THEN OX1 = LENGTH - (AB + AA1) / 2 ELSE OX1 = 0
                    FX2 = R4:     FX3 = LENGTH - FX1 - FX2
        FY3 = S:       FY2 = WID - S:  FY1 = WID - FY3 - FY2
        SZ1 = 0:       SZ2 = H1:      SZ3 = HIGHT - SZ1 - SZ2
        SY3 = WID - AREAS1 / H1:      SY2 = WID - SY3:      SY1 = WID - SY2 - SY3
                    OX2 = (AB + AA1) / 2:           OX3 = LENGTH - OX1 - OX2
        OZ1 = 0:       OZ2 = H:       OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC9:  '------------ SUBROUTINE FOR CASE # 9 ------------------------------------------

        PP = S + ((X1 - Z) * (S1 - S) * TAN(CITA) / (F * COS(WALT) * TAN(AZMD)))
        AA = R * (WID - PP) / (S1 - S)
        AREAS = M * (PP - S) + (2 * M - AA) * (WID - PP) / 2
        R4 = AREAS / (WID - S)
        R1 = R - ((S1 - WID) * R / (S1 - S))
        H = (S1 - WID) * TAN(CITA)
        AB = (P4 - WID) * TAN(AZMD):          R2 = AB - M + AA
        AC = AA * H / R1:                     H1 = 2 * AC / 3
        IF SPOS = 1 THEN FX1 = LENGTH - R4 - (X1 - Z) / 2 ELSE FX1 = (X1   Z) / 2
        IF SPOS = 1 THEN OX1 = LENGTH - AB + R2 / 2 ELSE OX1 = 0
                    FX2 = R4:     FX3 = LENGTH - FX1 - FX2
```

```
        FY3 = S:      FY2 = WID - S:  FY1 = WID - FY3 - FY2
        SZ1 = 0:      SZ2 = H1:     SZ3 = HIGHT - SZ1 - SZ2
        SY3 = WID - (AA * H * (WID - PP)) / (2 * R1 * H1):       SY2 = WID - SY3:     SY1 = WID - SY2 - SY3
                  OX2 = AB - R2 / 2:                      OX3 = LENGTH - OX1 - OX2
        OZ1 = 0.      OZ2 = H:      OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC10: '------------ SUBROUTINE FOR CASE # 10 ----------------------------------------

        R1 = R - ((S1 - WID) * R / (S1 - S)):         R2 = M + X1 - Z
        R4 = (2 * R2 - R1) / 2
        AB = (P4 - WID) * TAN(AZMD):            AA1 = X1 + M - Z - R1
        H1 = 2 * Z1 / 3:        H = (S1 - WID) * TAN(CITA)
        AREAS = .5 * (S1 - S) * Z1 - .5 * (S1 - WID) * H + .5 * F * (S - P2) * COS(WALT - CITA) / COS(CITA)
        IF SPOS = 1 THEN FX1 = LENGTH - R4 ELSE FX1 = 0
        IF SPOS = 1 THEN OX1 = LENGTH - (AB + AA1) / 2 ELSE OX1 = 0
                  FX2 = R4:     FX3 = LENGTH - FX1 - FX2
        FY3 = S:      FY2 = WID - S:  FY1 = WID - FY3 - FY2
        SZ1 = 0:      SZ2 = H1:     SZ3 = HIGHT - SZ1 - SZ2
        SY3 = WID - (AREAS / H1):     SY2 = WID - SY3:     SY1 = WID - SY2 - SY3
                  OX2 = (AB + AA1) / 2:          OX3 = LENGTH - OX1 - OX2
        OZ1 = 0:      OZ2 = H:      OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC11: '------------ SUBROUTINE FOR CASE # 11 ----------------------------------------

        H = (S1 - WID) * TAN(CITA)
        R1 = R - ((S1 - WID) * R / (S1 - S))
        R2 = M + X1 - Z
        AB = (P1 - WID) * TAN(AZMD):    AA = M + X1 - Z - R1
        H1 = H * (M - AA) / (M - AA + AB)
        AREAS = (M + AA - AB) * H / 2 + .5 * (M - AA + AB) * H - .5 * (M - AA) * H1
        R5 = AREAS / H:         HH = (S - P2) * COS(WALT - CITA) / COS(CITA)
        HHH = SQR((H1 - Z2) ^ 2 + (WID - P2) ^ 2)
        H2 = 2 * H1 / 3:        R4 = (R1 + R2) / 2
        AREAS1 = .5 * (WID - S) * H1 + .5 * HHH * HH
        IF SPOS = 1 THEN FX1 = LENGTH - R4 ELSE FX1 = 0
        IF SPOS = 1 THEN OX1 = LENGTH - R5 - AB / 2 ELSE OX1 = AB / 2
                  FX2 = R4:     FX3 = LENGTH - FX1 - FX2
        FY3 = S:      FY2 = WID - S:  FY1 = WID - FY3 - FY2
        SZ1 = 0:      SZ2 = H2:     SZ3 = HIGHT - SZ1 - SZ2
        SY3 = WID - AREAS1 / H2:      SY2 = WID - SY3:     SY1 = WID - SY2 - SY3
                  OX2 = R5:     OX3 = LENGTH - OX1 - OX2
        OZ1 = 0:      OZ2 = H:      OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC12: '------------ SUBROUTINE FOR CASE # 12 ----------------------------------------

        H = (S1 - WID) * TAN(CITA)
        AB = (P4 - WID) * TAN(AZMD)
        H1 = (S - WID) * TAN(CITA)
        AREAS = (H - H1) * (WID - P1) + .5 * F * (P1 - P2) * COS(WALT - CITA) / COS(CITA)
        H2 = (Z1 + Z2 + H1 + H) / 4:          H3 = H - H1
        AA = (P3 - WID) * TAN(AZMD):          R5 = (AB + AA) / 2
        .F SPOS = 1 THEN OX1 = LENGTH - R5 ELSE OX1 = 0
        FX1 = 0:      FX2 = 0.      FX3 = LENGTH - FX1 - FX2
        FY3 = 0:      FY2 = 0·      FY1 = WID - FY3 - FY2
        SZ1 = H2 - H3 / 2:           SZ2 = H3:      SZ3 = HIGHT - SZ1 - SZ2
        SY3 = WID - (AREAS / H3).     SY2 = WID - SY3:     SY1 = WID - SY2 - SY3
                  OX2 = R5:     OX3 = LENGTH - OX1 - OX2
        OZ1 = H1:     OZ2 = H - H1:  OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC13: '------------ SUBROUTINE FOR CASE # 13 ----------------------------------------

        IF WALT = 0 THEN HH = F:      HH1 = F:      GOTO EE2:
        HH = (WID - P2) / SIN(WALT).   HH1 = (WID - P3) / SIN(WALT)
EE2:    H = HH * COS(WALT) + Z2.      H1 = HH1 * COS(WALT) + Z3
        AREAS = (HH + HH1) * (P3 - P2) * COS(WALT - CITA) / (2 * COS(CITA))
        H2 = (H + H1 + Z3 + Z2) / 4·   MZZ = WID - (P3 + P2) / 2
        H3 = AREAS / MZZ
        AB = (P4 - WID) * TAN(AZMD)             AA = (P1 - WID) * TAN(AZMD)
        H6 = (S1 - WID) * TAN(CITA):          H7 = (S - WID) * TAN(CITA)
        AREAS1 = (AB - AA) * H6 + (H6 + H) * AA / 2 - (H6 + H1) * AB / 2
        MZZ1 = H6 - (H + H1) / 2
        IF SPOS = 1 THEN OX1 = LENGTH - AREAS1 / MZZ1 - AA / 2 ELSE OX1 = 0
        FX1 = 0:      FX2 = 0:      FX3 = LENGTH - FX1 - FX2
        FY3 = 0:      FY2 = 0:      FY1 = WID - FY3 - FY2
        SZ1 = H2 - H3 / 2:           SZ2 = H3.      SZ3 = HIGHT - SZ1 - SZ2
        SY3 = WID - MZZ:             SY2 = WID - SY3:     SY1 = WID - SY2 - SY3
```

```
            OX2 = AREAS1 / MZZ1:            OX3 = LENGTH - OX1 - OX2
        OZ1 = H6 - MZZ1:            OZ2 = MZZ1:         OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC14:  '―――――― SUBROUTINE awR CASE # 14 ――――――――――――――――――――――――――――――

        IF WALT = 0 THEN HH1 = F ELSE HH1 = (WID - P3) / SIN(WALT)
        H1 = HH1 * COS(WALT) + Z3
        H = (S1 - WID) * TAN(CITA):     HH = (Z1 - H) / SIN(CITA)
        H2 = HH * COS(CITA - WALT)
        AREAS = F * H2 + ((F + HH1) / 2) * ((P3 - P2) * COS(WALT - CITA) / COS(CITA) - H2)
        H3 = (Z3 + Z2 + Z1 + H1 + H) / 5:       H4 = (Z1 - Z2 - Z3 + H) / 2
        MZZ = AREAS / H4:               AB = (P4 - WID) * TAN(AZMD)
        AREAS1 = .5 * (H - H1) * AB:    R5 = AREAS1 / (H - H1)
        IF SPOS = 1 THEN OX1 = LENGTH - R5 ELSE OX1 = 0
        FX1 = 0:        FX2 = 0:        FX3 = LENGTH - FX1 - FX2
        FY3 = 0:        FY2 = 0:        FY1 = WID - FY3 - FY2
        SZ1 = H3 - H4 / 2:              SZ2 = H4:               SZ3 = HIGHT - SZ1 - SZ2
        SY3 = WID - MZZ:                SY2 = WID - SY3:        SY1 = WID - SY2 - SY3
                OX2 = LENGTH - OX1:                     OX3 = LENGTH - OX1 - OX2
        OZ1 = H1:       OZ2 = H - H1:   OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

EC15:  '―――――― SUBROUTINE FOR CASE # 15 ――――――――――――――――――――――――――――――

        H = (S1 - WID) * TAN(CITA):     H1 = (S - WID) * TAN(CITA)
        H3 = (WID - P2) * TAN(PI / 2 - WALT) + (WID - P2) * TAN(CITA) + H1
        AB = (P3 - WID) * TAN(AZMD)
        AREAS = M * (H - H3) + (M + AB) * (H3 - H1) / 2
        AA = (P1 - WID) * TAN(AZMD):    R4 = AA / 2:    R5 = AREAS / (H - H1)
        AREAS1 = (H3 - H1) * (WID - P2) / 2
        MZZ = 2 * (WID - P2) / 3:       H4 = AREAS1 / MZZ:      H5 = (H1 + Z2 + H3) / 3
        IF SPOS = 1 THEN OX1 = LENGTH - R5 - R4 ELSE OX1 = R4
        FX1 = 0:        FX2 = 0:        FX3 = LENGTH - FX1 - FX2
        FY3 = 0:        FY2 = 0:        FY1 = WID - FY3 - FY2
        SZ1 = H5 - H4 / 2:              SZ2 = H4:               SZ3 = HIGHT - SZ1 - SZ2
        SY3 = WID - MZZ:                SY2 = WID - SY3:        SY1 = WID - SY2 - SY3
                OX2 = R5:       OX3 = LENGTH - OX1 - OX2
        OZ1 = H1:       OZ2 = H - H1:   OZ3 = HIGHT - OZ2 - OZ1
        GOTO 14200

13400  '****************** SUBROUTINE TO CALCULATE BEAM ILLUMINANCE ON A SURFACE  ( KLUX ) **************************
       '        req: JULN, SALT, SKCO, INANG

        IF SKCO = 3 THEN EDH = 0: EDV = 0: RETURN
        IF SKCO = 1 THEN C = .21 ELSE C = .8
        EXTS = 127.5 * (1 + .034 * COS(2 * PI * (JULN - 2) / 365))
        EDN = EXTS * EXP(-1 * C / SIN(SALT))
        EDH = EDN * SIN(SALT):          EDH(SKCO) = EDH
        IF AZMD > (PI / 2) THEN EDV = 0 ELSE EDV = EDN * COS(INANG)
        RETURN

13450  '****************** SUBROUTINE TO CALCULATE SURFACE LUMINANCE ACCORDING TO BEAM RADIATION ************************
       '        req:

      '―――――――――― LUMINANCE ON OPPOSITE WALL ――――――――――――――――――――――――――――――――
        SLOP = PI / 2:          AZMD = ABS(SAZMS - RODA(I, W1, 3)):     IF AZMD > PI THEN AZMD = (2 * PI) - AZMD
        GOSUB 5200:             GOSUB 13400:
        LOPP = EDV * RODA(I, W1, 4) * WIDA(I, I2, 7) * WIDA(I, I2, 8)

      '―――――――――― LUMINANCE ON SIDE WALL ――――――――――――――――――――――――――――――――――――
        SLOP = PI / 2:          IF AZMD <= PI / 2 THEN AZMD = (PI / 2) - AZMD ELSE AZMD = AZMD - (PI / 2)
        GOSUB 5200:             GOSUB 13400:
        LSID = EDV * RODA(I, W1, 4) * WIDA(I, I2, 7) * WIDA(I, I2, 8)

      '―――――――――― LUMINANCE ON FLOOR ――――――――――――――――――――――――――――――――――――――――
        LFLO = EDH * RODA(I, 5, 4) * WIDA(I, I2, 7) * WIDA(I, I2, 8)
        RETURN

13550  '****************** SUBROUTINE TO CALCULATE ILLUMINANCE IN A POINT FROM TILTED SOURCE **************************

        AAA = SQR(H ^ 2 + Q ^ 2 - 2 * Q * H * COS(SLOP1))
        BBB = SQR(W ^ 2 + Q ^ 2 * (SIN(SLOP1)) ^ 2)
        RA1 = ATN((H - Q * COS(SLOP1)) / BBB) + ATN((Q * COS(SLOP1)) / BBB)
        ETILT = (L / 2) * (ATN(W / Q) + (H * COS(SLOP1) - Q) * ATN(W / AAA) / AAA + (W * COS(SLOP1) * RA1 / BBB))
        RETURN

13600  '****************** SUBROUTINE TO CALCULATE ILLUMINANCE IN A POINT FROM PARALLEL SOURCE *************************
       '        req: Q, H, W  (dimensions),  L (luminance)
```

```
        RA1 = SQR(H ^ 2 + Q ^ 2)
        RA2 = SQR(W ^ 2 + Q ^ 2)
        EPAR = (L / 2) * ((H / RA1) * ATN(W / RA1) + (W / RA2) * ATN(H / RA2))
        RETURN

13650   '****************** SUBROUTINE TO CALCULATE ILLUMINANCE IN A POINT FROM PERPENDICULAR SOURCE *********************
      '   req: Q, H, W  (dimensions),  L (luminance)

        RA1 = SQR(H ^ 2 + Q ^ 2)
        EPRN = (L / 2) * (ATN(W / Q) - (Q / RA1) * ATN(W / RA1))
        RETURN

13700   '****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13600  AND GET FINAL RESULTS ************
      '   req: F, M, N, S, R, U  (as in drawings)      { M+N < U  or  U < N }

        Q = F:     IF U > M + N THEN H = U - N ELSE H = M + N - U:  W = S + R:    GOSUB 13600:   LUM1 = EPAR:     '(ABCD
        Q = F:     IF U > M + N THEN H = U - N ELSE H = M + N - U:  W = S:      GOSUB 13600:   LUM2 = EPAR:     '(BC)
        Q = F:     IF U > M + N THEN H = U - M - N ELSE H = N - U
                              W = S + R:    GOSUB 13600:   LUM3 = EPAR:        '(CD)
        Q = F:     IF U > M + N THEN H = U - M - N ELSE H = N - U
                              W = S:        GOSUB 13600:   LUM4 = EPAR:        '(C)
        ELL = LUM1 - LUM2 - LUM3 + LUM4
        RETURN

13710   '****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13600  AND GET FINAL RESULTS ************
      '   req: F, M, N, K, J, U  (as in drawings),  L (Luminance)       { M+N < U  or  U < N }

        Q = F:     IF U > M + N THEN H = U - N ELSE H = M + N - U:
             W = K:       GOSUB 13600:   LUM1 = EPAR:        '(AD)
        Q = F:     IF U > M + N THEN H = U - N ELSE H = M + N - U:
             W = J:       GOSUB 13600.   LUM2 = EPAR:        '(A1D1)
        Q = F:     IF U > M + N THEN H = U - M - N ELSE H = N - U
                        W = K:       GOSUB 13600:   LUM3 = EPAR:        '(D)
        Q = F:     IF U > M + N THEN H = U - M - N ELSE H = N - U
                        W = J:       GOSUB 13600:   LUM4 = EPAR:        '(D1)
        ELL = LUM1 + LUM2 - LUM3 - LUM4
        RETURN

13720   '****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13600  AND GET FINAL RESULTS ************
      '   req: F, M, N, S, R  (as in drawings)      { N < U < M+N }

        Q = F:     H = M + N - U:     W = S + R:             GOSUB 13600:   LUM1 = EPAR:     '(AB)
        Q = F:     H = U - N:         W = S + R:             GOSUB 13600:   LUM2 = EPAR:     '(CD)
        Q = F:     H = M + N - U:     W = S:                 GOSUB 13600:   LUM3 = EPAR:     '(B)
        Q = F:     H = U - N:         W = S:                 GOSUB 13600:   LUM4 = EPAR:     '(C)
        ELL = LUM1 + LUM2 - LUM3 - LUM4
        RETURN

13730   '****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13600  AND GET FINAL RESULTS ************
      '   req: F, M, N, K, J  (as in drawings),  L (Luminance)      { N < U < M+N }

        Q = F:     H = M + N - U:  W = K:      GOSUB 13600:   LUM1 = EPAR:     '(A)
        Q = F:     H = M + N - U:  W = J:      GOSUB 13600:   LUM2 = EPAR:     '(B)
        Q = F:     H = U - N:      W = J:      GOSUB 13600:   LUM3 = EPAR:     '(C)
        Q = F:     H = U - N:      W = K:      GOSUB 13600:   LUM4 = EPAR:     '(D)
        ELL = LUM1 + LUM2 + LUM3 + LUM4
        RETURN

13750   '****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13650  AND GET FINAL RESULTS ************
      '   req: F, M, N, S, R  (as in drawings)

        IF N < 0 THEN N = 0
        Q = F:     H = M + N:    W = S + R:       GOSUB 13650:   LUM1 = EPRN:     '(ABCD   look Murdock Ex:2.13)
        Q = F:     H = M + N:    W = S:           GOSUB 13650:   LUM2 = EPRN:     '(BC)
        Q = F:     H = N:        W = S + R:       GOSUB 13650:   LUM3 = EPRN:     '(CD)
        Q = F:     H = N:        W = S:           GOSUB 13650:   LUM4 = EPRN:     '(C)
        ELL = LUM1 - LUM2 - LUM3 + LUM4
        RETURN

13760   '****************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13650  AND GET FINAL RESULTS ************
      '   req: F, M, N, K, J  (as in drawings),  L (Luminance)

        IF N < 0 THEN N = 0
        Q = F:     H = M + N:    W = K:           GOSUB 13650.   LUM1 = EPRN:     '(AD)
        Q = F.     H = M + N:    W = J:           GOSUB 13650:   LUM2 = EPRN:     '(A1D1)
        Q = F:     H = N:        W = K:           GOSUB 13650:   LUM3 = EPRN:     '(D)
        Q = F.     H = N:        W = J:           GOSUB 13650:   LUM4 = EPRN:     '(D1)
        ELL = LUM1 + LUM2 - LUM3 - LUM4
        RETURN
```

```
13770 '***************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13550  AND GET FINAL RESULTS ***********
      '   req: F, M, N, K, J  (as in drawings),  L (Luminance)

          IF N < 0 THEN N = 0
          Q = F:      H = M + N.    W = K:              GOSUB 13550:   LUM1 = ETILT:        '(AD)
          Q = F:      H = M + N:    W = J:              GOSUB 13550:   LUM2 = ETILT:        '(A1D1)
          Q = F:      H = N:      W = K:                GOSUB 13550:   LUM3 = ETILT:        '(D)
          Q = F:      H = N:      W = J:                GOSUB 13550:   LUM4 = ETILT:        '(D1)
          ELL = LUM1 + LUM2 - LUM3 - LUM4
          RETURN

13775 '***************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13550  AND GET FINAL RESULTS ***********
      '   req: F, M, N, S, R  (as in drawings)

          IF N < 0 THEN N = 0
          Q = F:      H = M + N:    W = S + R:          GOSUB 135⁵0:   LUM1 = ETILT:        '(ABCD   look Murdock Ex:2.13)
          Q = F:      H = M + N:    W = S:              GOSUB 13550:   LUM2 = ETILT:        '(BC)
          Q = F:      H = N:      W = S + R:            GOSUB 13550:   LUM3 = ETILT:        '(CD)
          Q = F:      H = N:      W = S:                GOSUB 13550:   LUM4 = ETILT:        '(C)
          ELL = LUM1 - LUM2 - LUM3 + LUM4
          RETURN

13780 '***************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13550  AND GET FINAL RESULTS ***********
      '   req: EE, U1 , M, N, K, J  (as in drawings),  L (Luminance)

          IF N < 0 THEN N = 0
          Q = U1:     H = EE - N:    W = K:             GOSUB 13550:   LUM1 = ETILT:        '(AD)
          Q = U1:     H = EE - N:    W = J:             GOSUB 13550:   LUM2 = ETILT:        '(A1D1)
          Q = U1:     H = EE - N - M: W = K:            GOSUB 13550:   LUM3 = ETILT:        '(D)
          Q = U1:     H = EE - N - M: W = J:            GOSUB 13550:   LUM4 = ETILT:        '(D1)
          ELL = LUM1 + LUM2 - LUM3 - LUM4
          RETURN

13785 '***************** SUBROUTINE TO DETERMINE THE VARIABLES FOR SUBROUTINES  13550  AND GET FINAL RESULTS ***********
      '   req: EE, U1, M, N, S, R  (as in drawings)

          IF N < 0 THEN N = 0
          Q = U1:     H = EE - N:    W = S + R:         GOSUB 13550:   LUM1 = ETILT:        '(ABCD   look Murdock Ex:2.13)
          Q = U1:     H = EE - N:    W = S:             GOSUB 13550:   LUM2 = ETILT:        '(BC)
          Q = U1:     H = EE - N - M: W = S + R:     GOSUB 13550·   LUM3 = ETILT:      '(CD)
          Q = U1:     H = EE - N - M: W = S:            GOSUB 13550    LUM4 = ETILT·        '(C)
          ELL = LUM1 - LUM2 - LUM3 + LUM4
          RETURN

13800 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON OPPOSITE WALL (1) FROM SUN SPOT *************************
13810 '--------------- FROM FLOOR ------------------------------------------------------------------------------
          F = HIGHT / 2:       M = FY2:       N = FY1:       Z = LENGTH / 2·       L = LFLO
RRE1:     IF Z < FX1 OR Z > FX1 + FX2 THEN GOTO ER1: ELSE GOTO ER2:
ER1:      R = FX2
          IF FX1 > FX3 THEN S = Z - R - FX3 ELSE S = Z - R - FX1
          GOSUB 13750. RETURN
ER2:      K = Z - FX1:   J = FX2 - K:        GOSUB 13760:   RETURN
13820 '--------------- FROM SIDE WALL --------------------------------------------------------------------------
          F = LENGTH / 2:      M = SY2:       N = SY1·       Z = HIGHT / 2.        L = LSID
RRE2:     IF Z > SZ2 + SZ3 OR Z < SZ3 THEN GOTO ER3: ELSE GOTO ER4:
ER3:      R = SZ2
          IF SZ1 > SZ3 THEN S = Z - R - SZ3 ELSE S = Z - R - SZ1
          GOSUB 13750: RETURN
ER4:      K = Z - SZ1:   J = SZ2 - K:        GOSUB 13760:   RETURN

13802 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON OPPOSITE WALL (2) FROM SUN SPOT **********************
13812 '--------------- FROM FLOOR ------------------------------------------------------------------------------
          F = HIGHT / 2:       M = FY2:       N = FY1:       L = LFLO: Z = LENGTH / 6.          GOTO RRE1:
13822 '--------------- FROM SIDE WALL --------------------------------------------------------------------------
          M = SY2:       N = SY1:       Z = HIGHT / 2.       L = LSID:       F = LENGTH / 6·        GOTO RRE2·
13804 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON OPPOSITE WALL (3) FROM SUN SPOT **********************
13814 '--------------- FROM FLOOR ------------------------------------------------------------------------------
          F = HIGHT / 2:       M = FY2:       N = FY1:       L = LFLO· Z = 5 * LENGTH / 6          GOTO RRE1:
13824 '--------------- FROM SIDE WALL --------------------------------------------------------------------------
          M = SY2.       N = SY1:       Z = HIGHT / 2:       L = LSID. F = 5 * LENGTH / 6          GOTO RRE2:

13850 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON FLOOR (1) FROM SUN SPOT *****************************
13860 '--------------- FROM OPPOSITE WALL ----------------------------------------------------------------------
          F = WID / 2:         M = OZ2:       N = OZ1:       Z = LENGTH / 2        L = LOPP
RRE3:     IF Z < OX1 OR Z > OX1 + OX2 THEN GOTO ER5: ELSE GOTO ER6·
ER5:      R = OX2
          IF OX1 > OX3 THEN S = Z - R - OX3 ELSE S = Z - R - OX1
          GOSUB 13750: RETURN
ER6:      K = Z - OX1:   J = OX2 - K:        GOSUB 13760:   RETURN
```

```
13870 '————————— FROM SIDE WALL ————————————————————
        F = LENGTH / 2:     M = SZ2:     N = SZ1:     Z = WID / 2:     L = LSID
RRE4:   IF Z > SY2 + SY3 OR Z < SY3 THEN GOTO ER7: ELSE GOTO ER8:
ER7:    R = SY2
        IF SY1 > SY3 THEN S = Z - R - SY3 ELSE S = Z - R - SY1
        GOSUB 13750: RETURN
ER8:    K = Z - SY1:  J = SY2 - K:     GOSUB 13760:  RETURN

13852 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON FLOOR (2) FROM SUN SPOT *****************************
13862 '————————— FROM OPPOSITE WALL ————————————————————
        M = OZ2:     N = OZ1.     Z = LENGTH / 2:     L = LOPP: F = WID / 6:     GOTO RRE3:
13872 '————————— FROM SIDE WALL ————————————————————
        F = LENGTH / 2:     M = SZ2:     N = SZ1:     L = LSID: Z = WID / 6:     GOTO RRE4:
13854 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON FLOOR (3) FROM SUN SPOT *****************************
13864 '————————— FROM OPPOSITE WALL ————————————————————
        M = OZ2:     N = OZ1:     Z = LENGTH / 2:     L = LOPP: F = 5 * WID / 6:     GOTO RRE3:
13874 '————————— FROM SIDE WALL ————————————————————
        F = LENGTH / 2:     M = SZ2:     N = SZ1:     L = LSID: Z = 5 * WID / 6:     GOTO RRE4:

13900 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL (1) FROM SUN SPOT *********************
13910 '————————— FROM OPPOSITE WALL ————————————————————
        F = WID / 2:     M = OX2:     N = OX1:     Z = HIGHT / 2:     L = LOPP
RRE5:   IF Z < OZ1 OR Z < OZ3 THEN GOTO ER9: ELSE GOTO ER10:
ER9:    R = OZ2
        IF OZ1 > OZ3 THEN S = Z - R - OZ3 ELSE S = Z - R - OZ1
        GOSUB 13750: RETURN
ER10:   K = Z - OZ1:  J = OZ2 - K:     GOSUB 13760:  RETURN
13920 '————————— FROM FLOOR ————————————————————
        F = HIGHT / 2:     M = FX2:     N = FX1:     Z = WID / 2:     L = LFLO
RRE6:   IF Z > FY2 + FY3 OR Z < FY3 THEN GOTO ER11: ELSE GOTO ER12:
ER11:   R = FY2
        IF FY1 > FY3 THEN S = Z - R - FY3 ELSE S = Z - R - FY1
        GOSUB 13750: RETURN
ER12:   K = Z - FY1:  J = FY2 - K:     GOSUB 13760:  RETURN
13930 '————————— FROM RIGHT SIDE WALL ————————————————————
        F = LENGTH:     M = SZ2:     N = SZ1:     Z = WID / 2:  U = HIGHT / 2:     L = LSID
RRE7:   IF Z > SY2 + SY3 OR Z < SY3 THEN GOTO ER13: ELSE GOTO ER14:
ER13:     R = SY2
        IF SY1 > SY3 THEN S = Z - R - SY3 ELSE S = Z - R - SY1
        IF U > N AND U < M + N THEN GOSUB 13720: RETURN
        GOSUB 13750: RETURN
ER14:   K = Z - SY1:  J = SY2 - K:
        IF U > N AND U < M + N THEN GOSUB 13730: RETURN
        GOSUB 13760:  RETURN

13902 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL (2) FROM SUN SPOT *********************
13912 '————————— FROM OPPOSITE WALL ————————————————————
        M = OX2:     N = OX1:     Z = HIGHT / 2:     L = LOPP: F = WID / 6:     GOTO RRE5:
13922 '————————— FROM FLOOR ————————————————————
        F = HIGHT / 2:     M = FX2:     N = FX1:     L = LFLO: Z = WID / 6:     GOTO RRE6:
13932 '————————— FROM RIGHT SIDE WALL ————————————————————
        F = LENGTH:     M = SZ2:     N = SZ1:     U = HIGHT / 2:     L = LSID: Z = WID / 6:  GOTO RRE7:
13904 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON LEFT SIDE WALL (3) FROM SUN SPOT *********************
13914 '————————— FROM OPPOSITE WALL ————————————————————
        M = OX2:     N = OX1:     Z = HIGHT / 2.     L = LOPP: F = 5 * WID / 6:     GOTO RRE5:
13924 '————————— FROM FLOOR ————————————————————
        F = HIGHT / 2:     M = FX2.     N = FX1:     L = LFLO: Z = 5 * WID / 6:     GOTO RRE6:
13934 '————————— FROM RIGHT SIDE WALL ————————————————————
        F = LENGTH:     M = SZ2:     N = SZ1·     U = HIGHT / 2:   L = LSID: Z = 5 * WID / 6: GOTO RRE7:

13950 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE (1) WALL FROM SUN SPOT *********************
13960 '————————— FROM OPPOSITE WALL ————————————————————
        F = WID / 2:     M = OX2.     N = OX3.     Z = HIGHT / 2:     L = LOPP
RRE8.   IF Z < OZ1 OR Z < OZ3 THEN GOTO ER9. ELSE GOTO ER10:
13970 '————————— FROM FLOOR ————————————————————
        F = HIGHT / 2:     M = FX2:     N = FX3·     Z = WID / 2:     L = LFLO
RRE9.   IF Z > FY2 + FY3 OR Z < FY3 THEN GOTO ER11: ELSE GOTO ER12:
13980 '————————— FROM LEFT SIDE WALL ————————————————————
        F = LENGTH·     M = SZ2     N = SZ1·     Z = WID / 2:  U = HIGHT / 2:     L = LSID
RRE10·  IF Z > SY2 + SY3 OR Z < SY3 THEN GOTO ER13· ELSE GOTO ER14·

13952 '***************** SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE (2) WALL FROM SUN SPOT *********************
13962 '————————— FROM OPPOSITE WALL ————————————————————
        M = OX2·     N = OX3.     Z = HIGHT / 2·     L = LOPP: F = WID / 6:     GOTO RRE8:
13972 '————————— FROM FLOOR ————————————————————
        F = HIGHT / 2:     M = FX2:     N = FX3:     L = LFLO: Z = WID / 6:     GOTO RRE9:
13982 '————————— FROM LEFT SIDE WALL ————————————————————
        F = LENGTH·     M = SZ2:     N = SZ1:     U = HIGHT / 2:     L = LSID: Z = WID / 6:  GOTO RRE10:
```

```
13954 '****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON RIGHT SIDE (3) WALL FROM SUN SPOT ************************
13964 '---------------- FROM OPPOSITE WALL ------------------------------------------------
      M = OX2:     N = OX3:      Z = HIGHT / 2:       L = LOPP: F = 5 * WID / 6:        GOTO RRE8:
13974 '---------------- FROM FLOOR ------------------------------------------------
      F = HIGHT / 2:      M = FX2:      N = FX3:      L = LFLO: Z = 5 * WID / 6:        GOTO RRE9:
13984 '---------------- FROM LEFT SIDE WALL ------------------------------------------------
      F = LENGTH:       M = SZ2:       N = SZ1:       U = HIGHT / 2: L = LSID: Z = 5 * WID / 6:   GOTO RRE10.

14000 '****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON CIELING (1) FROM SUN SPOT ****************************
14010 '---------------- FROM OPPOSITE WALL ------------------------------------------------
      F = WID / 2:       M = OZ2:       N = OZ3:       Z = LENGTH / 2:       L = LOPP
RRE11:  IF Z < OX1 OR Z > OX1 + OX2 THEN GOTO ER5: ELSE GOTO ER6:
14020 '---------------- FROM SIDE WALL ------------------------------------------------
      F = LENGTH / 2:      M = SZ2:       N = SZ3:       Z = WID / 2:       L = LSID
RRE12:  IF Z > SY3 + SY2 OR Z < SY3 THEN GOTO ER7: ELSE GOTO ER8:
14030 '---------------- FROM FLOOR ------------------------------------------------
      F = HIGHT:       M = FY2:       N = FY1:       Z = LENGHT / 2:     U = WID / 2:       L = LFLO
RRE13:  IF Z < FX1 OR Z > FX1 + FX2 THEN GOTO ER15: ELSE GOTO ER16:
ER15:     R = FX2
        IF FX1 > FX3 THEN S = Z - R - FX3 ELSE S = Z - R - FX1
        IF U > N AND U < M + N THEN GOSUB 13720: RETURN
        GOSUB 13750: RETURN
ER16:   K = Z - FX1:   J = FX2 - K:
        IF U > N AND U < M + N THEN GOSUB 13730: RETURN
        GOSUB 13760:   RETURN

14002 '****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON CIELING (2) FROM SUN SPOT ****************************
14012 '---------------- FROM OPPOSITE WALL ------------------------------------------------
      M = OZ2:      N = OZ3:       Z = LENGTH / 2:       L = LOPP: F = WID / 6:        GOTO RRE11:
14022 '---------------- FROM SIDE WALL ------------------------------------------------
      F = LENGTH / 2:      M = SZ2:       N = SZ3:       L = LSID: Z = WID / 6:        GOTO RRE12:
14032 '---------------- FROM FLOOR ------------------------------------------------
      F = HIGHT:       M = FY2:      N = FY1:      Z = LENGHT / 2:     L = LFLO: U = WID / 6:    GOTO RRE13:
14004 '****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON CIELING (3) FROM SUN SPOT ****************************
14014 '---------------- FROM OPPOSITE WALL ------------------------------------------------
      M = OZ2:      N = OZ3:       Z = LENGTH / 2:       L = LOPP: F = 5 * WID / 6:       GOTO RRE11:
14024 '---------------- FROM SIDE WALL ------------------------------------------------
      F = LENGTH / 2:      M = SZ2:       N = SZ3:       L = LSID: Z = 5 * WID / 6:       GOTO RRE12:
14034 '---------------- FROM FLOOR ------------------------------------------------
      F = HIGHT:       M = FY2:      N = FY1.      Z = LENGHT / 2:     L = LFLO: U = 5 * WID / 6·  GOTO RRE13.

14050 '****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON WINDOW WALL (1) FROM SUN SPOT ************************
14060 '---------------- FROM FLOOR ------------------------------------------------
      F = HIGHT / 2:       M = FY2:       N = FY3:       Z = LENGTH / 2        L = LFLO
RRE14:  IF Z < FX1 OR Z > FX1 + FX2 THEN GOTO ER1: ELSE GOTO ER2:
14070 '---------------- FROM SIDE WALL ------------------------------------------------
      F = LENGTH / 2:      M = SY2:       N = SY3:       Z = HIGHT / 2:       L = LSID
RRE15:  IF Z < SZ1 OR Z < SZ3 THEN GOTO ER3.  ELSE GOTO ER4:
14080 '---------------- FROM OPPOSITE WALL ------------------------------------------------
      F = WID:       M = OZ2:       N = OZ1:       Z = LENGHT / 2:     U = HIGHT / 2:       L = LOPP
RRE16:  IF Z < FX1 OR Z > FX1 + FX2 THEN GOTO ER17: ELSE GOTO ER18:
ER17:     R = OX2
        IF OX1 > OX3 THEN S = Z - R - OX3 ELSE S = Z - R - OX1
        IF U > N AND U < M + N THEN GOSUB 13720. RETURN
        GOSUB 13750: RETURN
ER18:   K = Z - OX1:   J = OX2 - K:
        IF U > N AND U < M + N THEN GOSUB 13730. RETURN
        GOSUB 13760:   RETURN

14052 '****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON WINDOW WALL (2) FROM SUN SPOT ************************
14062 '---------------- FROM FLOOR ------------------------------------------------
      F = HIGHT / 2·      M = FY2:      N = FY3:      L = LFLO: Z = LENGTH / 6:         GOTO RRE14·
14072 '---------------- FROM SIDE WALL ------------------------------------------------
      M = SY2       N = SY3:       Z = HIGHT / 2:       L = LSID: F = LENGTH / 6:        GOTO RRE15:
14082 '---------------- FROM OPPOSITE WALL ------------------------------------------------
      F = WID.       M = OZ2:       N = OZ1:       U = HIGHT / 2:     L = LOPP: Z = LENGTH / 6: GOTO RRE16·
14054 '****************** SUBROUTINE TO CALCULATE ILLUMINANCE ON WINDOW WALL (3) FROM SUN SPOT ************************
14064 '---------------- FROM FLOOR ------------------------------------------------
      F = HIGHT / 2:      M = FY2:       N = FY3:       L = LFLO: Z = 5 * LENGTH / 6:        GOTO RRE14.
14074 '---------------- FROM SIDE WALL ------------------------------------------------
      M = SY2:       N = SY3:       Z = HIGHT / 2.      L = LSID: F = 5 * LENGTH / 6:       GOTO RRE15.
14084 '---------------- FROM OPPOSITE WALL ------------------------------------------------
      F = WID.       M = OZ2:       N = OZ1:       U = HIGHT / 2:  L = LOPP. Z = 5 * LENGTH / 6:   GOTO RRE16

14200 '****************** SUBROUTINE TO CALCULATE ROOM SURFACES ILLUMINANCE DUE TO SUN SPOT ONLY ************************

      ARR = WIDA(I, I2, 4) * WIDA(I, I2, 5)
      CALL OVERHANG(SALT, SAZMS, ARR, FI)
      GOSUB 13810: E1 = ELL:       GOSUB 13820
             REFS(I, W3, SKCO) = REFS(I, W3, SKCO) + (ELL + E1) * FI:                'OPPOSITE WALL (1) ILLUMINANCE
```

```
GOSUB 13812: E1 = ELL:        GOSUB 13822
          REFS(I, W3 + 6, SKCO) = REFS(I, W3 + 6, SKCO) + (ELL + E1) * FI:          'OPPOSITE WALL (2) ILLUMINANCE
GOSUB 13814: E1 = ELL:        GOSUB 13824
          REFS(I, W3 + 12, SKCO) = REFS(I, W3 + 12, SKCO) + (ELL + E1) * FI:        'OPPOSITE WALL (3) ILLUMINANCE

GOSUB 13860: E1 = ELL:        GOSUB 13870
          REFS(I, 5, SKCO) = REFS(I, 5, SKCO) + (ELL + E1) * FI:                    'FLOOR (1) ILLUMINANCE
GOSUB 13862: E1 = ELL:        GOSUB 13872
          REFS(I, 5 + 6, SKCO) = REFS(I, 5 + 6, SKCO) + (ELL + E1) * FI:            'FLOOR (2) ILLUMINANCE
GOSUB 13864: E1 = ELL:        GOSUB 13874
          REFS(I, 5 + 12, SKCO) = REFS(I, 5 + 12, SKCO) + (ELL + E1) * FI:          'FLOOR (3) ILLUMINANCE

GOSUB 13910: E1 = ELL:        GOSUB 13920: E1 = ELL + E1
          IF SPOS = 1 THEN GOSUB 13930:  E1 = ELL + E1
          REFS(I, W4, SKCO) = REFS(I, W4, SKCO) + E1 * FI:                          'LEFT SIDE WALL (1) ILLUMINANCE
GOSUB 13912: E1 = ELL:        GOSUB 13922: E1 = ELL + E1
          IF SPOS = 1 THEN GOSUB 13932:  E1 = ELL + E1
          REFS(I, W4 + 6, SKCO) = REFS(I, W4 + 6, SKCO) + E1 * FI:                  'LEFT SIDE WALL (2) ILLUMINANCE
GOSUB 13914: E1 = ELL:        GOSUB 13924: E1 = ELL + E1
          IF SPOS = 1 THEN GOSUB 13934:  E1 = ELL + E1
          REFS(I, W4 + 12, SKCO) = REFS(I, W4 + 12, SKCO) + E1 * FI:                'LEFT SIDE WALL (3) ILLUMINANCE

GOSUB 13960: E1 = ELL:        GOSUB 13970: E1 = ELL + E1
          IF SPOS = 2 THEN GOSUB 13980:  E1 = ELL + E1
          REFS(I, W2, SKCO) = REFS(I, W2, SKCO) + E1 * FI:                          'RIGHT SIDE WALL (1) ILLUMINANCE
GOSUB 13962: E1 = ELL:        GOSUB 13972: E1 = ELL + E1
          IF SPOS = 2 THEN GOSUB 13982:  E1 = ELL + E1
          REFS(I, W2 + 6, SKCO) = REFS(I, W2 + 6, SKCO) + E1 * FI:                  'RIGHT SIDE WALL (2) ILLUMINANCE
GOSUB 13964: E1 = ELL:        GOSUB 13974: E1 = ELL + E1
          IF SPOS = 2 THEN GOSUB 13984:  E1 = ELL + E1
          REFS(I, W2 + 12, SKCO) = REFS(I, W2 + 12, SKCO) + E1 * FI:                'RIGHT SIDE WALL (3) ILLUMINANCE

GOSUB 14010: E1 = ELL:        GOSUB 14020: E1 = ELL + E1
          GOSUB 14030: REFS(I, 6, SKCO) = REFS(I, 6, SKCO) + (ELL + E1) * FI:              'CIELING (1) ILLUMINANCE
GOSUB 14012: E1 = ELL:        GOSUB 14022: E1 = ELL + E1
          GOSUB 14032: REFS(I, 6 + 6, SKCO) = REFS(I, 6 + 6, SKCO) + (ELL + E1) * FI:      'CIELING (2) ILLUMINANCE
GOSUB 14014: E1 = ELL:        GOSUB 14024: E1 = ELL + E1
          GOSUB 14034: REFS(I, 6 + 12, SKCO) = REFS(I, 6 + 12, SKCO) + (ELL + E1) * FI:    'CIELING (3) ILLUMINANCE

GOSUB 14060: E1 = ELL:        GOSUB 14070: E1 = ELL + E1
          GOSUB 14080: REFS(I, W1, SKCO) = REFS(I, W1, SKCO) + (ELL + E1) * FI:            'WINDOW WALL (1) ILLUMINANCE
GOSUB 14062: E1 = ELL:        GOSUB 14072: E1 = ELL + E1
          GOSUB 14082: REFS(I, W1 + 6, SKCO) = REFS(I, W1 + 6, SKCO) + (ELL + E1) * FI:    'WINDOW WALL (2) ILLUMINANCE
GOSUB 14064: E1 = ELL:        GOSUB 14074: E1 = ELL + E1
          GOSUB 14084: REFS(I, W1 + 12, SKCO) = REFS(I, W1 + 12, SKCO) + (ELL + E1) * FI:  'WINDOW WALL (3) ILLUMINANCE
RETURN

'***************************************************************************************************************
'*********************** FILE READING AND WRITING **************************************************************
'***************************************************************************************************************

20100 '********** SUBROUTINE TO READ EXISTING FILE FROM DRIVE *************************************************

OPEN FIL$ FOR INPUT AS #1
INPUT #1, CITY.N$
FOR I10 = 1 TO 7: FOR I11 = 1 TO 12: INPUT #1, BIDA(I10, I11): NEXT I11: NEXT I10
NSP = BIDA(1, 1):      GOSUB 155
FOR I10 = 1 TO NSP. FOR I11 = 1 TO 8:  FOR I12 = 1 TO 7:  INPUT #1, RODA(I10, I11, I12): NEXT I12: NEXT I11
          FOR I11 = 1 TO 10: FOR I12 = 1 TO 20: INPUT #1, WIDA(I10, I11, I12): NEXT I12: NEXT I11
          FOR I11 = 1 TO 10: FOR I12 = 1 TO 5:  INPUT #1, REFP(I10, I11, I12): NEXT I12: NEXT I11
 NEXT I10
   CLOSE
   RETURN

SUB OVERHANG (SALT, SAZMS, ARR, FI)
5500 '****************** SUBROUTINE TO CALCULATE THE SHADOW AREA OVER A WINDOW ************************************
'       ( required : SAZMN, SALT, window Azimuth & Altitude )

        AA = WIDA(I, I2.      PP = WIDA(I, I2, 10):    BB = WIDA(I, I2, 11):    D1 = WIDA(I, I2, 12)
        PD = WIDA(I, I2, 13)   BD = WIDA(I, I2, 14):    CD = WIDA(I, I2, 15):    AD = WIDA(I, I2, 16)
        L = WIDA(I, I2, 4):    H = WIDA(I, I2, 5)
        AZMD = ABS(SAZMS - WIDA(I, I2, 19))
        IF AZMD > PI THEN AZMD = (2 * PI) - AZMD
        ALTD = SALT - WIDA(I, I2, 6)
        IF ALTD < 0 THEN AREV = 0. GOTO B6600:
'----------------------------------------------Overhang Calculation-------------------------------------------------
5510    IF PP <= 0 THEN GOTO B6600
        T = PP * TAN(ALTD) / COS(AZMD)
        M = PP * TAN(AZMD)
        AB = BB * T / M
```

```
         DE = (H + AA) * M / T
         UG = (L + BB) * T / M
         IF T <= AA THEN GOTO B5750:
         IF AB <= AA THEN GOTO B5690:
         IF DE <= BB THEN GOTO B5730:
         IF M <= BB THEN GOTO B5770
         IF BE <= (L + BB) THEN GOTO B5740:
         IF M <= (L + BB) THEN GOTO B5810:  ELSE GOTO B5830:
B5690:   IF UG <= AA THEN GOTO B5750:
         IF DE <= (L + BB) THEN GOTO B5720:
         IF M <= (L + BB) THEN GOTO B5780:  ELSE GOTO B5800:
B5720:   IF T <= (H + AA) THEN GOTO B5780:  ELSE GOTO B5790:
B5730:   IF T < (H + AA) THEN GOTO B5770:  ELSE GOTO B5760:
B5740:   IF T <= (H + AA) THEN GOTO B5810:  ELSE GOTO B5820:
B5750:   AREO = 0: GOTO B5840:
B5760:   AREO = H * L: GOTO B6960:
B5770:   AREO = L * (T - AA): GOTO B5890:
B5780:   AREO = (T - AA) * (L + BB - M * (1 + AA / T) / 2): GOTO B5885:
B5790:   AREO = H * (L - ((AA + H / 2) * M / T) + BB): GOTO B6600:
B5800:   AREO = (((L + BB) * T / M) - AA) ^ 2 * M / (2 * T): GOTO B6600:
B5810:   AREO = ((T - AA) * L) - ((M - BB) ^ 2 * T / (2 * M)): GOTO B5885:
B5820:   AREO = (H * L) - (((H + AA) * M / T) - BB) ^ 2 * T / (2 * M): GOTO B6600:
B5830:   AERO = L * (((BB + L / 2) * T / M) - AA): GOTO B6600:
B5840:   IF (T + D1) <= AA THEN GOTO B5900:
B5850:   IF M <= BB THEN GOTO B5870:
B5860:   IF M < (L + BB) THEN GOTO B5880:  ELSE GOTO B5900:
B5870:   IF (T + D1) < (H + AA) THEN GOTO B5920:  ELSE GOTO B5910:
B5880:   IF (T + D1) <= (H + AA) THEN GOTO B5940:  ELSE GOTO B5930:
B5885:   IF (T + D1) <= (H + AA) THEN GOTO B5970:  ELSE GOTO B5980:
B5890:   IF (T + D1) <= (H + AA) THEN GOTO B5950:  ELSE GOTO B5960:
B5900:   AREV = 0: GOTO B6600:
B5910:   AREV = H * L: GOTO B6960:
B5920:   AREV = L * (T + D1 - AA): GOTO B6600:
B5930:   AREV = H * (L + BB - M): GOTO B6600:
B5940:   AREV = (T + D1 - AA) * (L + BB - M): GOTO B6600:
B5950:   AREV = L * D1: GOTO B6600:
B5960:   AREV = L * (H + AA - T): GOTO B6960:
B5970:   AREV = (L + BB - M) * D1: GOTO B6600:
B5980:   AREV = (L + BB - M) * (H + AA - T): GOTO B6600:
       '-------------------------FIN CALCULATIONS----------------------------------------
B6600:   IF PD <= 0 THEN GOTO B6960:
         TD = PD * TAN(ALTD) / COS(AZMD)
B6610:   MD = PD * TAN(AZMD)
B6620:   ABD = BD * TD / MD
B6630:   UGD = (L + BD) * TD / MD
B6640:   DED = (H + AD) * MD / TD
B6650:   DJD = CD * MD / TD
B6660:   IF MD <= BD THEN GOTO B6780:
B6670:   IF ABD < AD THEN GOTO B6710:
B6680:   IF DED <= BD THEN GOTO B6780:
B6690:   IF UGD <= (H + AD) THEN GOTO B6770:
B6700:   IF TD <= (H + AD) THEN GOTO B6810:  ELSE GOTO B6830:
B6710:   IF UGD <= AD THEN GOTO B6750:
B6720:   IF TD <= AD THEN GOTO B6800:
B6730:   IF UGD <= (H + AD) THEN GOTO B6760:
B6740:   IF TD <= (H + AD) THEN GOTO B6840:  ELSE GOTO B6860:
B6750:   IF MD <= (L + BD) THEN GOTO B6800:  ELSE GOTO B6790:
B6760:   IF MD <= (L + BD) THEN GOTO B6840:  ELSE GOTO B6850:
B6770:   IF MD <= (L + BD) THEN GOTO B6810:  ELSE GOTO B6820:
B6780:   AREF = 0: GOTO B6960:
B6790:   AREF = H * L: GOTO B6900:
B6800:   AREF = H * (MD - BD): GOTO B6880:
B6810:   AREF = (MD - BD) * ((H + AD) - (TD * (1 + BD / MD) / 2)): GOTO B6880:
B6820:   AREF = L * (H - ((BD + L / 2) * TD / MD) + AD): GOTO B6900:
B6830:   AREF = (((H + AD) * MD / TD) - BD) ^ 2 * TD / (2 * MD): GOTO B6900:
B6840:   AREF = ((MD + BD) * H) - ((TD - AD) ^ 2 * MD / (2 * TD)): GOTO B6880:
B6850:   AREF = (H * L) - (((L + BD) * TD / MD) - AD) ^ 2 * MD / (2 * TD): GOTO B6900:
B6860:   AREF = H * (((AD + H / 2) * MD / TD) - BD): GOTO B6900:
       '-------------------------CALCULATE SHORT FIN--------------------------------------
B6880:   IF DJD <= BD THEN GOTO B6920:
B6890:   IF DJD <= MD THEN GOTO B6950.  ELSE GOTO B6930:
B6900:   IF DJD <= BD THEN GOTO B6920:
B6910:   IF DJD <= (L + BD) THEN GOTO B6950:  ELSE GOTO B6940:
B6920:   ARES = 0: GOTO B6960:
B6930:   ARES = -1 * (MD - BD) * (CD - (TD * (1 + BD / MD) / 2)): GOTO B6960:
B6940:   ARES = -1 * L * (CD - ((BD + L / 2) * TD / MD)): GOTO B6960:
B6950:   ARES = -1 * (CD - (BD * TD / MD)) ^ 2 * MD / (2 * TD): GOTO B6960:
B6960:   ARET = AREO + AREV + AREF + ARES
         FI = (ARR - ARET) / ARR
5520     END SUB
```

Solar Energy Routine

```
'**************************************************************************************************************
'*************** SOLAR ENERGY CALCULATIONS ********************************************************************
'**************************************************************************************************************
DECLARE SUB OVERHANG (SALT, SAZMS, ARR, FI)

        OPEN "DATAS.AAA" FOR INPUT AS #2:     INPUT #2, SALT, SAZMS, JULN, MON, STT, FIL$: CLOSE
        DIM SHARED BIDA(7, 12), FI, I, I2:     GOSUB 20100
        PI = 3.141592653#: GOTO 160
155     DIM SHARED RODA(NSP, 8, 16), WIDA(NSP, 10, 20), WEATH(13, 7), WASE(NSP, 12), RADT(NSP, 24), GTT(NSP, 4, 24): RETURN
160     GOSUB 40005: FOR I = 1 TO BIDA(1, 1):    RADT(I, STT) = 0:     FOR I2 = 1 TO RODA(I, 7, 3): GOSUB 5000: GOSUB 5400
        NEXT I2:     NEXT I
        OPEN "RESS.AAA" FOR OUTPUT AS #1: FOR I = 1 TO NSP: PRINT #1, RADT(I, STT): FOR I4 = 1 TO 4: PRINT #1, GTT(I, I4, STT): NEXT I4: NEXT
        I:CLOSE
        END

5000  '******* SUBROUTINE FOR HOURLY SOLAR RADIATION ON A SURFACE ************************************************
    '   req: (LAT, WEATH(13,3)(km), JULN)

        IDN = AAJU / EXP(BBJU / SIN(SALT))
        GOSUB 5210
        IB = IDN * COS(INANG)
        ID = CCJU * IDN * (1 + SIN(WIDA(I, I2, 6))) / 2
        IG = IDN * (CCJU + SIN(SALT)) * RODA(I, 7, 7) * (1 - SIN(WIDA(I, I2, 6))) / 2
          RETURN

5400  '*********** SUBROUTINE TO CALCULATE HOURLY RADIATION THROUGH A WINDOW ******
    '   (req:- control function)

        I4 = WIDA(I, I2, 1)
        AZMD = ABS(SAZMS - WIDA(I, I2, 19))
        IF AZMD > PI THEN AZMD = (2 * PI) - AZMD
        IF COS(AZMD) <= .0001 THEN IB = 0:    '(no sun enter through this window)
        SRAZM = WIDA(I, I2, 19):      ARR = WIDA(I, I2, 4) * WIDA(I, I2, 5)
        GOSUB 5210
        RA16 = INANG:        GOSUB 6000:         TRB = RA23
        RA16 = (90 - .5788 * SLOP * 180 / PI + .002693 * (SLOP * 180 / PI) ^ 2) * PI / 180:  GOSUB 6000: TRG = RA23
        RA16 = (59.68 - .1388 * SLOP * 180 / PI + .001497 * (SLOP * 180 / PI) ^ 2) * PI / 180:   GOSUB 6000:  TRD = RA23
        CALL OVERHANG(SALT, SAZMS, ARR, FI)
        ALLFA = (1 - RODA(I, 7, 1)) / (1 - RODA(I, 7, 1) + RODA(I, 7, 1) * TRD * ARR / (RODA(I, 5, 7) + RODA(I, 7, 2) + RODA(I, 6, 7)))
        RADT = 3600 * ARR * ALLFA * (FI * TRB * IB + TRD * ID + TRG * IG)
        RADT(I, STT) = RADT(I, STT) + RADT:  GTT(I, I4, STT) = 3600 * (IB + ID + IG)
        RETURN:             '( Without Control Function ,  With Absorbtion )

5210  '******* SUBROUTINE TO CALCULATE INCIDENT ANGLE ON A SURFACE **********************
    '   (req:- SLOP,  SALT)

        SLOP = PI / 2 - WIDA(I, I2, 6):
        RA9 = COS(SALT) * COS(AZMD) * SIN(SLOP) + SIN(SALT) * COS(SLOP)
        INANG = ATN(SQR(1 - RA9 ^ 2) / RA9)
        IF INANG < 0 THEN INANG = INANG + PI
          RETURN

6000  '********** SUBROUTINE TO CALCULATE TRANSMITTANCE THROUGH GLASS ******************

        KK = WIDA(I, I2, 17):         LL = WIDA(I, I2, 18)
        RA17 = SIN(RA16) / 1.526
        RA18 = ATN(RA17 / SQR(1 - (RA17) ^ 2))
        IF RA18 < 0 THEN RA18 = RA18 + PI
        RA19 = (SIN(RA18 - RA16)) ^ 2 / (SIN(RA18 + RA16)) ^ 2
        RA20 = (TAN(RA18 - RA16)) ^ 2 / (TAN(RA18 + RA16)) ^ 2
        RA21 = .5 * ((1 - RA20) / (1 + RA20) + (1 - RA19) / (1 + RA19))
        RA22 = EXP(-1 * KK * LL / COS(RA18))
        RA23 = RA21 * RA22
        RETURN

20100 '*********** SUBROUTINE TO READ EXISTING FILE FROM DRIVE ********************************
        OPEN FIL$ FOR INPUT AS #1
        INPUT #1, CITY.N$
        FOR I10 = 1 TO 7: FOR I11 = 1 TO 12: INPUT #1, BIDA(I10, I11): NEXT I11: NEXT I10
        NSP = BIDA(1, 1):      GOSUB 155
        FOR I10 = 1 TO NSP: FOR I11 = 1 TO 8:  FOR I12 = 1 TO 7:  INPUT #1, RODA(I10, I11, I12): NEXT I12: NEXT I11
                FOR I11 = 1 TO 10: FOR I12 = 1 TO 20: INPUT #1, WIDA(I10, I11, I12): NEXT I12: NEXT I11
                FOR I11 = 1 TO 10: FOR I12 = 1 TO 5:  INPUT #1, REFP(I10, I11, I12): NEXT I12: NEXT I11
        NEXT I10
        CLOSE
```

```
        OPEN CITY.N$ FOR INPUT AS #1
        FOR I10 = 1 TO 13: FOR I11 = 1 TO 7: INPUT #1, WEATH(I10, I11): NEXT I11: NEXT I10: CLOSE
        LAT = WEATH(13, 1): LONGT = WEATH(13, 2): STM = WEATH(13, 4)
          RETURN

40005 '********************* SUBROUTINE TO CALCULATE THE JULIAN DAY ****************************
      '  req: I,I3,RODA( ),DAY.D%, MON

        ON MON GOSUB 40010, 40020, 40030, 40040, 40050, 40060, 40070, 40080, 40090, 40100, 40110, 40120:   RETURN
40010   AAJU = 1229: BBJU = .142: CCJU = .058: RETURN
40020   AAJU = 1214: BBJU = .144: CCJU = .06: RETURN
40030   AAJU = 1185: BBJU = .156: CCJU = .071: RETURN
40040   AAJU = 1135: BBJU = .18: CCJU = .097: RETURN
40050   AAJU = 1103: BBJU = .196: CCJU = .121: RETURN
40060   AAJU = 1088: BBJU = .205: CCJU = .134: RETURN
40070   AAJU = 1084: BBJU = .207: CCJU = .136: RETURN
40080   AAJU = 1107: BBJU = .201: CCJU = .122: RETURN
40090   AAJU = 1151: BBJU = .177: CCJU = .092: RETURN
40100   AAJU = 1192: BBJU = .16: CCJU = .073: RETURN
40110   AAJU = 1220: BBJU = .149: CCJU = .063: RETURN
40120   AAJ'    1233: BBJU = .142: CCJU = .057: RETURN
```

Program to Plot Curves (PLOT1)

```
' ****************** SUBROUTINE TO PLOT *****************
'          REQUIRED : M(NUMBER OF VARIABLES ON Y), YLAR, YMIN

           OPEN "RESUD.AAA" FOR INPUT AS #15: INPUT #15, RES$: CLOSE
           CLS
           XLAR = 12: XMIN = 1
           XTITLE$ = "MONTH": YTITLE$ = "Kw.hr (Thousands)"
           DIM X(12), Y(12, 6), LEG$(5)
           OPEN RES$ FOR INPUT AS #15:
           INPUT #15, M, TITLE$
           FOR NN = 1 TO 12: INPUT #15, X(NN): FOR MM = 1 TO M: INPUT #15, Y(NN, MM): Y(NN, MM) = Y(NN, MM) / 1000: NEXT MM
           NEXT NN: FOR MM = 1 TO M: INPUT #15, LEG$(MM): NEXT MM

           CLOSE

' *************** SUBROUTINE TO DETERMINE THE LOWEST AND HIGHEST VALUE OF Y ********
           YLAR = Y(1, 1): YMIN = Y(1, 1)
           FOR NN = 1 TO 12
           FOR MM = 1 TO M: IF Y(NN, MM) > YLAR THEN YLAR = Y(NN, MM)
                     IF Y(NN, MM) < YMIN THEN YMIN = Y(NN, MM)
           NEXT MM: NEXT NN


PLOT: REM ********   SUBROUTINE CURVE FITTING *************

      REM ************LARGEST AND SMALLEST X & Y OF DATA POINTS*********

      REM --XMAX=NEAREST NUMBER WITH A ZERO AT END (i.e. 30, OR 90, etc.)
           XMAX = INT(XLAR)


      REM --YMAX=NEAREST NUMBER WITH A ZERO AT END (i.e. 30, OR 90, etc.)
           YMAX = INT(YLAR + 1)
           YMIN = INT(YMIN)

2555 CLS : SCREEN 9: COLOR , 0

      REM *************** PLOT AXIES **********************************
           LINE (110, 49)-(560, 259), 6, BF
           LINE (110, 49)-(560, 259), 15, B

           FOR I = 259 TO 49 STEP -42: LINE (110, I)-(114, I), 15: NEXT I
           FOR I = 110 TO 560 STEP 40.9: LINE (I, 259)-(I, 255), 15: NEXT I


           COLOR 14: LOCATE 3, 15: PRINT YTITLE$
           LOCATE 19.5, 74: PRINT XTITLE$
           COLOR 7
           LET XDIST = XMAX - XMIN + 1
           LET YDIST = YMAX - YMIN
           LET I = 0
           FOR J = 4 TO 20 STEP 3
           LET YSC = (YMAX - (I * YDIST / 5))
           LOCATE J, 8, PRINT USING "###.##"; YSC
           I = I + 1: NEXT J
           LET L = 0
           FOR K = 14 TO 69 STEP 5
           LET XSC = (XMIN + (L * XDIST / 12))
           LOCATE 20 5, K: PRINT XSC
           L = L + 1: NEXT K

      REM *************PLOT DATA POINTS**********************************
           WINDOW (XMIN, YMIN)-(XMAX, YMAX)
           VIEW (110, 49)-(560, 259)

           JC = 1
           FOR JYJ = 1 TO M: JC = JC + 1
           FOR JJJ = 1 TO 12
           LINE (X(JJJ - 1), Y(JJJ - 1, JYJ))-(X(JJJ), Y(JJJ, JYJ)), JC
           CIRCLE (X(JJJ), Y(JJJ, JYJ)), XDIST / 300, JC
           NEXT JJJ: NEXT JYJ

'*************** TITLE AND LEGEND ******************************
           COLOR 6. LOCATE 3, 48· PRINT TITLE$
           JC = 2. FOR MM = 1 TO 2: COLOR JC
           LOCATE 22, 11 * MM ^ 2 + 1: PRINT LEG$(MM): JC = JC + 1
```

```
        NEXT MM
        FOR MM = 3 TO 4: COLOR JC
        LOCATE 23, 11 * (MM - 2) ^ 2 + 1: PRINT LEG$(MM): JC = JC + 1
        NEXT MM
        COLOR 7: LOCATE 1, 27: PRINT "--- Press <Esc> to exit ---"

OK:     REM

100     CHECK$ = INKEY$

        IF CHECK$ = CHR$(27) THEN GOTO QITE: ELSE GOTO 100

QITE:   ERASE X, Y: SCREEN 0: CLS

END
```

Lotus-123 Customized Program

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
                         Now you are in the LOTUS-123 program.
                    use the given menus to see the results in graphic form
                 You can plot the graphs and make any necessary changes
                               as normaly done in the LOTUS.
                          When you finish, just quit the lotus and you
                           will be back in your nice "DAYLIGHT" program.
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
{HOME}{BIGRIGHT 2}{WINDOWSOFF}{PANELOFF}
{GOTO}PATH~/fiLOTUS1.DAT~{home}
/fd
pathD:\QB45\AHM
~{HOME}/fnRESULT.AAA~
{menubranch bbb}


bbb    RADIATION          AVAILABLE_DAYLIGHT  REFERECES_ILLUMINANCE     BACK_TO_DAYLIGHT
       Plot graph for the monthly daily available solar radiation.    Plot graphs for the available daylight in different sky conditions.
       Plot graphs for the illuminance on the room sensors.  End LOTUS-123
       {branch rrr}       {menubranch aaa}           {menubranch lii}            {branch eee}

eee/qyy


AAA    CLEAR_CHART        PARTLY_CHART        OVERCAST_CHART     MAIN_MENU
       Show graph for the available daylight in clear sky condition      Show graph for the available daylight in partly cloudy sky condition
       Show graph for the available daylight in overcast sky condition   RADIATION     AVAILABLE_DAYLIGHT      REFEENCES_ILLUM.
       BACK_TO_DAYLIGHT
~/grgtlx{HOME}.{END}{DOWN}~        ~/grgtlx{HOME}.{END}{DOWN}~        ~/grgtlx{HOME}.{END}{DOWN}~        {menubra ich bbb}
a{HOME}{RIGHT}.{END}{DOWN}~        a{HOME}{RIGHT 4}.{END}{DOWN}~      a{HOME}{RIGHT 7}.{END}{DOWN}~
b{HOME}{RIGHT 2}.{END}{DOWN}~      b{HOME}{RIGHT 5}.{END}{DOWN}~      b{HOME}{RIGHT 8}.{END}{DOWN}~
c{HOME}{RIGHT 3}.{END}{DOWN}~      c{HOME}{RIGHT 6}.{END}{DOWN}~      c{HOME}{RIGHT 9}.{END}{DOWN}~
oclaSky Light Illum.~     oclaSky Light Illum.~     oclaSky Light Illum.~
lbSun Beam Illum.~        lbSun Beam Illum.~        lbSun Beam Illum.~
lcTotal Illum.~           lcTotal Illum.~           lcTotal Illum.~
tlClear Sky Available Daylight~      tlPartly Cloudy Sky Available Daylight~  tlOver Cast Sky Available Daylight~
txTime (Hours)~           txTime (Hours)~           txTime (Hours)~
tyIllumination (Klux)~    tyIllumination (Klux)~    tyIllumination (Klux)~
qvq{HOME}{BIGRIGHT 2}              qvq{HOME}{BIGRIGHT 2} qvq{HOME}{BIGRIGHT 2}
{menubranaaa}             {menubranch aaa}          {menubranch aaa}

rrr{HOME}/grgtlx{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
a{HOME}{RIGHT}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
octlHOURLY AVAILABLE SOLAR ENERGY~
txTIME (Hours)~
tyRADIATION (MegaJoules)~
qvq{HOME}{BIGRIGHT 2}
{menubranch bbb}

FFF{HOME}/grgtlx{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
a{HOME}{RIGHT}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
b{HOME}{RIGHT 2}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
c{HOME}{RIGHT 3}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
d{HOME}{RIGHT 4}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
e{HOME}{RIGHT 5}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
f{HOME}{RIGHT 6}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
octlILLUMINATION ON THE SENSORS~
tsFROM CLEAR SKY~
txTIME (Hours)~
tyILLUMINATION (Klux)~
laSENSOR#1~
lbSENSOR#2~
lcSENSOR#3~
ldSENSOR#4~
leSENSOR#5~
lfSENSOR#6~
qvq{HOME}{BIGRIGHT 2}
{menubranch lii}
```

```
GGG{HOME}/grgdx{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
a{HOME}{RIGHT}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
b{HOME}{RIGHT 2}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
c{HOME}{RIGHT 3}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
d{HOME}{RIGHT 4}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
e{HOME}{RIGHT 5}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
f{HOME}{RIGHT 6}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
octfILLUMINATION ON THE SENSORS~
tsFROM PARTLY CLOUDY SKY~
txTIME (Hours)~
tyILLUMINATION (Klux)~
laSENSOR#1~
lbSENSOR#2~
lcSENSOR#3~
ldSENSOR#4~
leSENSOR#5~
lfSENSOR#6~
qvq{HOME}2)
{menubranch iii}

HHH{HOME}/grgdx{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
a{HOME}{RIGHT}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
b{HOME}{RIGHT 2}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
c{HOME}{RIGHT 3}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
d{HOME}{RIGHT 4}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
e{HOME}{RIGHT 5}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
f{HOME}{RIGHT 6}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}{END}{DOWN}.{END}{DOWN}~
octfILLUMINATION ON THE SENSORS~
tsFROM OVERCAST SKY~
txTIME (Hours)~
tyILLUMINATION (Klux)~
laSENSOR#1~
lbSENSOR#2~
lcSENSOR#3~
ldSENSOR#4~
leSENSOR#5~
lfSENSOR#6~
qvq{HOME}{BIGRIGHT 2}
{menubranch iii}


iii          CLEAR_CHART        PARTLY_CHART        OVERCAST_CHART        MAIN_MENU
             Show graph for the sensors illuminance in clear sky condition      Show graph for the sensors illuminance in partly cloudy sky condition
             Show graph for the sensor illuminance in overcast sky condition      RADIATION      AVAILABLE_DAYLIGHT      REFEENCES_ILLUM.
             BACK_TO_DAYLIGHT
{branch fff} {branch ggg}       {branch hhh}        {menubranch bbb}
{BIGRIGHT ch
```