ENGR-UH 1001
Computer Programming for Engineers

Assignment 3 - Electrical Engineering Case Study ("Off-Grid" Solar Panel Installation Costs)

Muhammad Zunair Viqar
Mzv205
Fall 2019

Next Page ---->

a) Step 1: Problem Identification and Statement

The Objective is to write the program code for a software that calculates the cost of the installation of a flat-panel photovoltaic power system for residential use in Abu Dhabi, only considering the cost of material with simplifying assumptions. The software asks the user to input the daily energy requirements in kWh, cost of the solar panel per square meter, cost of power inverters and cost of batteries. It uses the daily energy requirements to calculate the total area of the solar panel, and then uses it to calculate the number of power inverters. With this information, the software calculates the total cost of the off-grid photovoltaic power system. The program allows the user to calculate the total cost of solar panels using a combination of different costs of the components.

b) Step 2: Gathering Information and I/O Description

In order to compute the cost of the installation of a flat-panel photovoltaic power system, we use the formulas and variables provided along with the input from the user and gather all missing information in one place.

- User Inputs and Constants:
    - Daily Energy Requirements (kWh) : DER
    - Solar Panel Cost (per square meter) : SP
    - Power Inverter Cost: PI
    - Batteries Cost: BATTERIES
    - Solar Cell Efficiency : SCF = 0.18
    - Power Inverter Efficiency : PWF = 0.90
    - Rows of the Matrix : ARRROW = 12
    - Columns of the Matrix : ARRCOLUMN = 24
    - Insolation Matrix : insolationmatrix[ARRROW][ARRCOLUMN]
    - Minimum Daily Insolation : insolDayMin
    - Maximum Insolation Hour : insolHourMax

- Given Data in Global Array
    - Seasonal Variation in the insolation from January to December
        S = seavar[ ] = {4.89, 5.52, 6.21, 6.88, 7.20, 6.76, 6.33, 6.39, 6.35, 6.04, 5.13, 4.68}

Since each day of the month is the same, the insolation is approximated on an hourly basis for each month using the formula :

$f_s(h) = 2 * S * \cos(0.2618(h-12))$ or
$f_s(h) = 2 * seavar[x] * \cos(0.2618(h-12))$

Where x is the month for which the insolation of each hour is calculated. The number of hours are 12 as solar energy is received during the 12 hours from dawn to dusk (i.e: 0700 hours to 1800 hours).

Next Page ---->

We use the above stated formula to fill in the Insolation Matrix from the 7th Row till the 18th row for each month (column) to achieve a matrix similar to the following:

0	0	0	0	0	0	2.5312	4.8899	6.9154	8.4697	9.4467	9.78	9.4467	8.4697	6.9154	4.8899	2.5312	-0.000035923	0	0	0	0	0	0
0	0	0	0	0	0	2.8573	5.5199	7.8064	9.5609	10.663	11.04	10.663	9.5609	7.8064	5.5199	2.8573	-0.000040552	0	0	0	0	0	0
0	0	0	0	0	0	3.2144	6.2099	8.7822	10.756	11.996	12.42	11.996	10.756	8.7822	6.2099	3.2144	-0.000045621	0	0	0	0	0	0
0	0	0	0	0	0	3.5613	6.8799	9.7297	11.916	13.291	13.76	13.291	11.916	9.7297	6.8799	3.5613	-0.000050543	0	0	0	0	0	0
0	0	0	0	0	0	3.7269	7.1999	10.182	12.470	13.909	14.4	13.909	12.470	10.182	7.1999	3.7269	-0.000052894	0	0	0	0	0	0
0	0	0	0	0	0	3.4991	6.7599	9.5600	11.708	13.059	13.52	13.059	11.708	9.5600	6.7599	3.4991	-0.000049661	0	0	0	0	0	0
0	0	0	0	0	0	3.2766	6.3299	8.9519	10.963	12.228	12.66	12.228	10.963	8.9519	6.3299	3.2766	-0.000046502	0	0	0	0	0	0
0	0	0	0	0	0	3.3076	6.3899	9.0368	11.067	12.344	12.78	12.344	11.067	9.0368	6.3899	3.3076	-0.000046943	0	0	0	0	0	0
0	0	0	0	0	0	3.2869	6.3499	8.9802	10.998	12.267	12.7	12.267	10.998	8.9802	6.3499	3.2869	-0.000046649	0	0	0	0	0	0
0	0	0	0	0	0	3.1264	6.0399	8.5418	10.461	11.668	12.08	11.668	10.461	8.5418	6.0399	3.1264	-0.000044372	0	0	0	0	0	0
0	0	0	0	0	0	2.6554	5.1299	7.2549	8.8854	9.9103	10.26	9.9103	8.8854	7.2549	5.1299	2.6554	-0.000037687	0	0	0	0	0	0
0	0	0	0	0	0	2.4225	4.6799	6.6185	8.1059	9.0410	9.36	9.0410	8.1059	6.6185	4.6799	2.4225	-0.000034381	0	0	0	0	0	0

This is calculated on Google Sheets at the following link using the same formula. (https://docs.google.com/spreadsheets/d/1TYmYeJBuSqZ-vduixPh0BgcGxvGPVxd5noSfigy2SkA/edit?usp=sharing)

Using this data, we calculate the total daily insolation for each month (from January to December) by adding all the values in each row:

74.28629951
83.85692705
94.33904293
104.5173294
109.3786005
102.6943527
96.16201961
97.07350795
96.46584905
91.75649264
77.93225286
71.09609033

From this, we store the least value of total daily insolation into insolDayMin:

- insolDayMin = 71.09

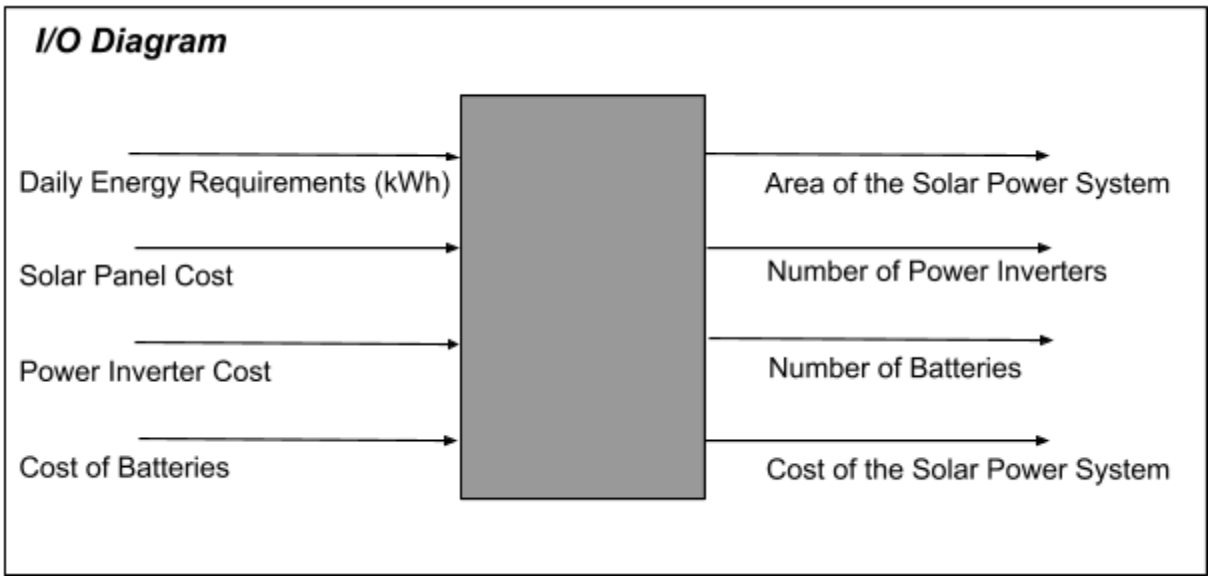
From the Matrix, we store the maximum value into insolHourMax:

- insolHourMax = 14.4

- Cost of the Solar Power System:
    - Area of the Solar panel : area = ((DER)/(SCF * insolDayMin))/PIF;
    - Number of Power Inverters : NOPI = (insolHourMax *(area/5))*(SCF/PIF) -round up this value
    - Total Cost of the Solar Power System = ((area * SP) + (NOPI *PI) + ((DER/4.8)*1200))
    (DER/4.8) calculates the Number of Batteries and rounds up the value to the nearest integer

Next Page ---->

The I/O diagram for this problem is illustrated below:



c) Step 3: Test Cases and algorithm

- Test Cases;

Test cases use sample input data that is used in manual calculations by the user to produce expected output data in order to verify the functionality of the program.For Test Cases, different ranges of data are used as follows;

Test Case 1:

(Normal Values)

DER = 20 , SP = 1000, PI = 1000, BATTERIES = 1200 are input by the user
Area = ((DER)/(SCF * insolDayMin))/PIF = ((20)/(0.18 * 71.09)/0.90 = (20/12.796)/0.90 = 1.56/0.90 = 1.736477907 = 1.736 = 1.74
NOPI = (insolHourMax *(area/5))*(SCF/PIF) = (14.4 * (1.736/5))*(0.18/0.90) = (14.4 * 0.34732533)*0.2 = 5.001 * 0.2 = 1.0002. Rounding it up makes it 2.
Number Of Batteries = DER / 4.8 = 20 / 4.8 = 4.167. Rounding it up makes it 5.
Cost = (Area * SP) + (NOPI * PI) + (Number Of Batteries * BATTERIES) = (1.73648*1000) + (2 * 1000) + (5 * 1200) = 1736.48 + 2000 + 6000 = 9736.48.

Next Page ---->

Test Case 2:

(Normal Values)

DER = 30 , SP = 1500, PI = 1000, BATTERIES = 1700 are input by the user
Area = ((DER)/(SCF * insolDayMin))/PIF = ((30)/(0.18 * 71.09)/0.90 = (30/12.796)/0.90 = 2.34/0.90 = 2.604 = 2.6
NOPI = (insolHourMax *(area/5))*(SCF/PIF) = (14.4 * (2.604/5))*(0.18/0.90) = (14.4 * 0.5208)*0.2 = 7.49952* 0.2 = 1.4999. Rounding it up makes it 2.
Number Of Batteries = DER / 4.8 = 30 / 4.8 = 6.25. Rounding it up makes it 7.
Cost = (Area * SP) + (NOPI * PI) + (Number Of Batteries * BATTERIES) = (2.60471*1500) + (2 * 1000) + (7 * 1700) = 3907.07 + 2000 + 11900 = 17807.1.

Test Case 3:

(Invalid Values)

DER = 26 , SP = 1300, PI = 2000
BATTERIES = -1200 are input by the user
Invalid value. Enter a positive value for the Cost of the Batteries.
Enter the Cost of the Batteries.

Test Case 4:

(Invalid Values)

DER = 22.5 , SP = 1280,
PI = - 1540 are input by the user
Invalid value. Enter a positive value for the Cost of the Power Inverters.
Enter the Cost of the Power Inverters.

Test Case 5:

(Invalid Values)

DER = 24
SP = - 1900 are input by the user
Invalid value. Enter a positive value for the Cost of the Solar Panels per square meter.
Enter the Cost of the Solar Panels per square meter.

Test Case 6:

(Invalid Values)

DER = -29.7 are input by the user
Invalid value. Enter a positive value for the Daily Energy Requirement.
Enter the daily Energy requirement in kWh.

Next Page ---->

Test Case 7:
(Extreme Values)

DER = 45 , SP = 10000, PI = 11500, BATTERIES = 10500 are input by the user
Area = ((DER)/(SCF * insolDayMin))/PIF = ((45)/(0.18 * 71.09)/0.90 = (45/12.796)/0.90 = 3.518/0.90 = 3.907 = 3.91
NOPI = (insolHourMax *(area/5))*(SCF/PIF) = (14.4 * (3.91/5))*(0.18/0.90) = (14.4 * 0.781)*0.2 = 11.25* 0.2 = 2.25. Rounding it up makes it 3.
Number Of Batteries = DER / 4.8 = 45 / 4.8 = 9.375. Rounding it up makes it 10.
Cost = (Area * SP) + (NOPI * PI) + (Number Of Batteries * BATTERIES) = (3.907075*10000) + (3 * 11500) + (10 * 10500) = 39070.75 + 34500 + 105000 = 178571.

Test Case 8: (Extreme Values)

DER = 10 , SP = 100, PI = 300, BATTERIES = 600 are input by the user
Area = ((DER)/(SCF * insolDayMin))/PIF = ((10)/(0.18 * 71.09)/0.90 = (10/12.796)/0.90 = 0.781/0.90 = 0.868 = 0.87 = 0.9
NOPI = (insolHourMax *(area/5))*(SCF/PIF) = (14.4 * (0.868/5))*(0.18/0.90) = (14.4 * 0.1736)*0.2 = 2.50 * 0.2 = 0.50. Rounding it up makes it 1.
Number Of Batteries = DER / 4.8 = 10 / 4.8 =2.0833. Rounding it up makes it 3.
Cost = (Area * SP) + (NOPI * PI) + (Number Of Batteries * BATTERIES) = (0.86823*100) + (1 * 300) + (3 * 600) = 86.823 + 300 + 2100 = 2186.82

Input					Output			
No.	DER	SP	PI	BATTERIES	Area	NOPI	No. of Batteries	Cost
1	20	1000	1000	1200	1.736	2	5	9736.48
2	30	1500	1000	1700	2.604	2	7	17807.1
3	26	1300	2000	-1200	Should prompt the user to input again			
4	22.5	1280	-1540	-	Should prompt the user to input again			
5	24	-1900	-	-	Should prompt the user to input again			
6	-29.7	-	-	-	Should prompt the user to input again			
7	45	10000	11500	10500	3.907	3	10	178571
8	10	100	300	600	0.868	1	3	2186.82

Next Page ---->

- Algorithm;

Assign 4.89,5.52,6.21,6.88,7.20,6.76,6.33,6.39,6.35,6.04,5.13,4.68 to seavar[]

Assign 0.18 to SCF
Assign 0.90 to PIF

Assign 12 to ARRROW
Assign 24 to ARRCOLUMN
Assign 12 to NOH

Main() Function
    Declare insolationmatrix as an array [ARRROW][ARRCOLUMN]
    Assign 0 to i
    Repeat while i is less than ARRROW
        Assign 0 to j
        Repeat while j is less than ARRCOLUMN
            Assign 0 to insolationmatrix[i][j]
            Increment j
        Increment i
    inputvaluefunction(DER, SP, PI, BATTERIES)

    Assign fillInInsolation(insolationmatrix, ARRROW) to insolDayMin

    Assign the address of insolHourMax to insolHourMaxPtr
    Assign searchMaxValue(insolationmatrix, ARRROW) to the content pointer by insolHourMaxPtr

    Assign ((DER)/(SCF * insolDayMin))/PIF to area
    Assign (insolHourMax *(area/5))*(SCF/PIF) to NOPI //Use Rounded value
    Assign (DER/4.8) to NumberOfBatteries //Use Rounded Value
    Assign ((area * SP) + (NOPI *PI) + (NumberOfBatteries)*BATTERIES) to cost

    Print "The total cost for the installation of a flat-panel photovoltaic (""solarcell"") power system for residential home use in Abu Dhabi, United Arab Emirates is $ " , cost
    Print "This takes into consideration the number of Power Inverters being "<< NOPI<< " , the number of batteries needed which are ", NumberOfBatteries, " and the area in m^2 of the solar panel ", area, newline

    Return 0

Next Page ---->

```
inputvaluefunction(DER, SP, PI, BATTERIES) //Pass by Reference
        Repeat
                Print “Enter the Enter the Daily Energy Requirement in kWh”, newline
                Read value into DER
                If DER is less than or equal to zero
                        Print “Invalid Value. Enter a Positive value for the Daily Energy Requirement",
        newline
        While DER is less than or equal to zero
        Repeat
                Print "Enter the Cost of the Solar Panels per square meter", newline
                Read value into SP
                If SP is less than or equal to zero
                        Print "Invalid Value. Enter a positive value for the Cost of the Solar Panels per
        square meter", newline
        While SP is less than or equal to zero
        Repeat
                Print "Enter the Cost of the Power Inverters", newline
                Read value into PI
                If PI is less than or equal to zero
                        Print "Invalid Value. Enter a Positive value for the Cost of the Power Inverters",
        newline
        While PI is less than or equal to zero
        Repeat
                Print "Enter the Cost of the Batteries", newline
                Read value into BatteriesCost
                If BatteriesCost is less than or equal to zero
                        Print "Invalid Value. Enter a Positive value for the Cost of the Batteries", newline
        While BatteriesCost is less than or equal to zero

fillInInsolation(InsMat[][ARRCOLUMN], ROWS)
        Assign 0 to i
        Repeat while i is less than ROWS
                Assign the address of InsMat[i][6] to pointer
                Assign dailyInsolation(pointer,NOH, i) to totalDailyInsolation[i]
                Increment i

        If totalDailyInsolation[0] is less than totaldailyinsolation[1]
                Assign totalDailyinsolation[0] to smallestDailyInsolation
        Otherwise
                Assign totalDailyInsolation[1] to smallestDailyInsolation
```

Next Page ---->

```
        Assign 0 to y
        Repeat while y is less than ROWS
                If totalDailyInsolation[y] is less than smallestDailyInsolation
                        Assign totalDailyInsolation[y] to smallestDailyInsolation
                Increment i

        Return smallestdailyInsolation

dailyInsolation(pointer, N, S)
        Assign 7 to h
        Assign 0 to i
        Repeat while i is less than N
                Assign (2*seavar[S]*(cos(0.2618 *(h-12)))) to content pointed to by pointer[i]
                Assign (pointer[i] + totalDailyInsolation) to totalDailyInsolation
                Increment h
                Increment i
        Return totalDailyInsolation

searchMaxValue(InsMat[][ARRCOLUMN],N)
        If InsMat[1][1] is greater than InsMat[1][2]
                Assign InsMat[1][1] to max
        Otherwise
                Assign InsMat[1][2] to max

        Assign 0 to a
        Repeat while a is less than ARRROW
                Assign 0 to b
                Repeat while b is less than ARRCOLUMN
                        If InsMat[a][b] is greater than max
                        Assign InsMat[a][b] to max
                        Increment b
                Increment a
        Return max
```

Next Page ---->

d) Step 4: Code or implementation

```
#include <iostream>
#include <cmath>
#include <string>

using namespace std;

//Global Array
double seavar[12] = {4.89,5.52,6.21,6.88,7.20,6.76,6.33,6.39,6.35,6.04,5.13,4.68}; // seavar = Seasonal Variation
//Global Variables
double SCF = 0.18; // Solar Cell Efficiency
double PIF = 0.90; // Power Inverter Efficiency
//Symbolic Constants
#define ARRROW 12 //Array Rows
#define ARRCOLUMN 24 //Array Columns
#define NOH 12 //Number Of Hours (Size of Array to be filled)
//function prototypes
void inputvaluefunction(double& DER,double& SP,double& PW,double& Batteriescost);
double fillInInsolation(double InsMat[][ARRCOLUMN], double ROWS);
double dailyInsolation(double* pointer, int N, int S);
double searchMaxValue(double InsMat[][ARRCOLUMN],int N);// Function to Search for the Maximum Value in Matrix
//Main Function
int main()
{
   double insolationmatrix[ARRROW][ARRCOLUMN];
   for(int i = 0; i < ARRROW; i++) // initialises the entire array to 0
      for(int j = 0; j < ARRCOLUMN; j++)
         insolationmatrix [i][j]= 0;
   double DER = 0,SP = 0 ,PI = 0,BatteriesCost = 0;
   inputvaluefunction(DER, SP, PI, BatteriesCost);

   double insolDayMin = 0;
   insolDayMin = fillInInsolation(insolationmatrix, ARRROW); //function call. stores the returned value from the function to the
   given variable

   double insolHourMax =0, *insolHourMaxPtr = &insolHourMax;
   *insolHourMaxPtr = searchMaxValue(insolationmatrix, ARRROW);//function call. stores the returned value to the variable
    pointed to by the pointer

   double area = 0;
   area = ((DER)/(SCF * insolDayMin))/PIF; // Formula to calculate the area of the solar panel

   int NOPI = 0; // Number of Power Inverters
   NOPI = ceil ((insolHourMax *(area/5))*(SCF/PIF));

   double NumberOfBatteries =0;
   NumberOfBatteries =(ceil(DER/4.8));

   double cost = 0;
```

Next Page ---->

```
    cost = ((area * SP) + (NOPI *PI) + (NumberOfBatteries)*BatteriesCost); //calculates the total cost of the solar power
    system. DER/4.8 calculate the number of batteries.
    cout << "The total cost for the installation of a flat-panel photovoltaic (""solarcell"") power system for residential home use
     in Abu Dhabi, United Arab Emirates is $ "<< cost;
    cout<< " This takes into consideration the number of Power Inverters being "<< NOPI<< ", the number of batteries needed
     which are "<< NumberOfBatteries<< " and the area in m^2 of the solar panel " << area <<endl;
    return 0;
}
void inputvaluefunction(double& DER,double& SP,double& PI,double& BatteriesCost)
{
    do{//loop to validate the input
    cout << "Enter the Daily Energy Requirement in kWh"<< endl;
    cin >> DER;
        if (DER<=0)
            cout<< "Invalid Value. Enter a positive value for the Daily Energy Requirement"<<endl;
    }while(DER<=0);

    do{//loop to validate the input
    cout << "Enter the Cost of the Solar Panels per square meter"<<endl;
    cin>> SP;
        if (SP<=0)
            cout<< "Invalid Value. Enter a positive value for the Cost of the Solar Panels per square meter"<<endl;
    }while(SP<=0);

    do{//loop to validate the input
    cout<< "Enter the Cost of the Power Inverters"<<endl;
    cin>> PI;
        if (PI<=0)
            cout<< "Invalid Value. Enter a positive value for the Cost of the Power Inverters"<<endl;
    }while(PI<=0);

    do{//loop to validate the input
     cout << "Enter the Cost of the Batteries"<<endl;
    cin>> BatteriesCost;
        if (BatteriesCost<=0)
            cout<< "Invalid Value. Enter a positive value for the Cost of the Batteries"<<endl;
    }while(BatteriesCost<=0);
}
//function to fill in the 2D array insolationmatrix by calling the dailyinsolation function to fill each row of the array
double fillInInsolation(double InsMat[][ARRCOLUMN], double ROWS)
{
    double totalDailyInsolation[ARRROW];
    double smallestDailyInsolation = 0;
    double* pointer;
    for (int i = 0;i < ROWS; i++){
        pointer = &InsMat[i][6]; //this statement inside a loop passes the starting address of each consecutive row in each run o
        the loop
        totalDailyInsolation[i] = dailyInsolation(pointer,NOH, i); // stores the total insolation of each day into an array
    }
    // to search for the smallest value in the array
```

Next Page ---->

```
    if (totalDailyInsolation[0]<totalDailyInsolation[1])
        smallestDailyInsolation=totalDailyInsolation[0];
    else
         smallestDailyInsolation=totalDailyInsolation[1];
    // the above statement executes a comparision between two consecutive values in the array and stores the lower value
    into the variable as a starting point
    for(int y = 0; y < ROWS; y++)
        if (totalDailyInsolation[y] < smallestDailyInsolation) // this follows the comparision for all other values in the array
            smallestDailyInsolation = totalDailyInsolation[y];
    return smallestDailyInsolation; // returns the minimum total daily insolation
}
//this function fills in the 2D array called insolationmatrix using its pointer and returns the total insolation from each day
double dailyInsolation(double* pointer, int N, int S){
    double totalDailyInsolation =0;
    int h = 7;
    for (int i=0; i < N; i++){
        *(pointer)= 2*seavar[S]*(cos(0.2618 *(h-12)));
        totalDailyInsolation += *(pointer);
        pointer++;
        h++;
    }
    return totalDailyInsolation;
}
// function to search the maximum value from the 2D array
double searchMaxValue(double InsMat[][ARRCOLUMN],int N)
{
    double max = 0;
    if (InsMat[1][1] > InsMat[1][2])
        max = InsMat[1][1];
    else
        max = InsMat[1][2];

    for (int a = 0; a < N; a++) // the following checks for the largest value in the array
        for (int b = 0; b < ARRCOLUMN; b++)
            if (InsMat[a][b]>max)
                max = InsMat[a][b];
    return max;
}

}
```

Next Page ---->

e) Step 5: Test and Verification

This step ensures if the final source code contains any errors or not. All the test cases in Step 3 are tried by running the program and the expected outcomes are compared with the actual outcomes.

Test Case 1:

```
Enter the Daily Energy Requirement in kWh
20
Enter the Cost of the Solar Panels per square meter
1000
Enter the Cost of the Power Inverters
1000
Enter the Cost of the Batteries
1200
The total cost for the installation of a flat-panel photovoltaic (solarcell) power system for
    residential home use in Abu Dhabi, United Arab Emirates is $ 9736.48 This takes into
    consideration the number of Power Inverters being 2, the number of batteries needed which
    are 5 and the area in m^2 of the solar panel 1.73648
Program ended with exit code: 0
```

Test Case 2:

```
Enter the Daily Energy Requirement in kWh
30
Enter the Cost of the Solar Panels per square meter
1500
Enter the Cost of the Power Inverters
1000
Enter the Cost of the Batteries
1700
The total cost for the installation of a flat-panel photovoltaic (solarcell) power system for
    residential home use in Abu Dhabi, United Arab Emirates is $ 17807.1 This takes into
    consideration the number of Power Inverters being 2, the number of batteries needed which
    are 7 and the area in m^2 of the solar panel 2.60472
Program ended with exit code: 0
```

Test Case 3:

```
Enter the Daily Energy Requirement in kWh
26
Enter the Cost of the Solar Panels per square meter
1300
Enter the Cost of the Power Inverters
2000
Enter the Cost of the Batteries
-1200
Invalid Value. Enter a positive value for the Cost of the Batteries
Enter the Cost of the Batteries
```

Next Page ---->

Test Case 4:

```
Enter the Daily Energy Requirement in kWh
22.5
Enter the Cost of the Solar Panels per square meter
1280
Enter the Cost of the Power Inverters
-1540
Invalid Value. Enter a positive value for the Cost of the Power Inverters
Enter the Cost of the Power Inverters
```

Test Case 5:

```
Enter the Daily Energy Requirement in kWh
24
Enter the Cost of the Solar Panels per square meter
-1900
Invalid Value. Enter a positive value for the Cost of the Solar Panels per square meter
Enter the Cost of the Solar Panels per square meter
```

Test Case 6:

```
Enter the Daily Energy Requirement in kWh
-29.7
Invalid Value. Enter a positive value for the Daily Energy Requirement
Enter the Daily Energy Requirement in kWh
```

Test Case 7:

```
Enter the Daily Energy Requirement in kWh
45
Enter the Cost of the Solar Panels per square meter
10000
Enter the Cost of the Power Inverters
11500
Enter the Cost of the Batteries
10500
The total cost for the installation of a flat-panel photovoltaic (solarcell) power system for
    residential home use in Abu Dhabi, United Arab Emirates is $ 178571 This takes into
    consideration the number of Power Inverters being 3, the number of batteries needed which
    are 10 and the area in m^2 of the solar panel 3.90708
Program ended with exit code: 0
```

Next Page ---->

Test Case 8:

```
Enter the Daily Energy Requirement in kWh
10
Enter the Cost of the Solar Panels per square meter
100
Enter the Cost of the Power Inverters
300
Enter the Cost of the Batteries
600
The total cost for the installation of a flat-panel photovoltaic (solarcell) power system for
    residential home use in Abu Dhabi, United Arab Emirates is $ 2186.82 This takes into
    consideration the number of Power Inverters being 1, the number of batteries needed which
    are 3 and the area in m^2 of the solar panel 0.868239
Program ended with exit code: 0
```

Next Page ---->

User Guide

This software can be used by Electrical Engineers for calculating the total cost for the installation of a flat-panel photovoltaic (“solar-cell”) power system for residential home use in Abu Dhabi UAE. the user has to only input the Daily Energy Requirement and the cost of Solar Panel per square meter, Power Inverters and Batteries. The software calculates, taking the units of measurements (or inputs) into account, the total cost by using data for the seasonal variation from January to December from https://www.sciencedirect.com/science/article/pii/S0038092X19306723). The Software then outputs the area of the Solar Panel, The Number of Power Inverters needed and the number of Batteries needed to store one full day’s energy use along with the total cost cost of these components added together.

Next Page ---->