NAME: SHUBHAM SHARMA
ROLL NO: 18i190002
MSC PHD (OR)

Exercise 1):

Q1) [R]

We have to formulate the mathematical model of the assignment problem given to us. The cost matrix is given

in the question.

Let **C=[c_{ij}]** be the cost matrix where c_{ij} is the setup cost of constructing the i_{th} factory to the j_{th} location. We have to minimize the sum of the costs of assigning the facilities to the locations.

<u>Defining the parameters</u>:-

defining a set S={1,2,3,4,5,6,7,8,9,10,11,12}.

(S is 12 because there 12 facilities and 12 locations)

Now we can see that basically our cost matrix **C** is defined on {S,S} , i.e C_{S*S}.

<u>Defining the decision variables:-</u>

We are defining 12*12=144 decision variables. i.e, we giving every machine a choice to go to every location.

Defining the variable X=[x_{ij}], where i, j € S.

Now each x_{ij} can have only two values,

x_{ij}={1, if factory 'i' is assigned to location 'j'.

0, if factory 'i' is not assigned to location 'j'.

i.e, eg x_{12} is the decision variable of assigning factory 1 to location 2.

Now , Defining the OBJECTIVE function:-

(As we have to minimize the cost of setting up)

Minimize cost: $\sum_{i,j=1}^{12}(\mathrm{Cij} * \mathrm{Xij})$

subject to constraints:-

1) As we know that one factory can go only one location: $\sum_{j=1}^{12} \mathrm{Xij} = 1$, for all i€S.
2) As we know that one location can have only one factory. $\sum_{i=1}^{12} \mathrm{Xij} = 1$, for all j€S.

Q2) [R]

(not mentioned but for AMPL model)

We have to formulate the mathematical model of the assignment problem given to us. The cost matrix is given

in the question.

Let **C=[c_{ij}]** be the cost matrix where c_{ij} is the setup cost of constructing the i_{th} factory to the j_{th} location. We have to minimize the sum of the costs of assigning the facilities to the locations.

Defining the parameters:-

Defining a parameter N.(which will tell us the number of factories or locations, eg for the above question N was 12)

defining a set S={1,2,………,N}.

Now we can see that basically our cost matrix **C** is defined on {S,S} , i.e C_{S*S}.

<u>Defining the decision variables:-</u>

We are defining $n*n=n^2$ decision variables. i.e, we giving every machine a choice to go to every location.

Defining the variable $X=[x_{ij}]$, where i, j € S.

Now each x_{ij} can have only two values,

x_{ij}={1, if factory 'i' is assigned to location 'j'.

0, if factory 'i' is not assigned to location 'j'.

i.e, eg x_{12} is the decision variable of assigning factory 1 to location 2.

Now , Defining the OBJECTIVE function:-

(As we have to minimize the cost of setting up)

Minimize cost: $\sum_{i,j=1}^{12}(\mathrm{Cij} * \mathrm{Xij})$

subject to constraints:-

3) As we know that one factory can go only one location: $\sum_{j=1}^{12} \mathrm{Xij} = 1$, for all i€S.
4) As we know that one location can have only one factory. $\sum_{i=1}^{12} \mathrm{Xij} = 1$, for all j€S.

Q4) [R]

We have solved the above problem using AMPL and the code files (ex1.mod,ex1.dat,ex1.run) are attached in the zipped file.

```
Console
AMPL
ampl: include ex1.run;
Gurobi 8.0.0: optimal solution; objective 198
17 simplex iterations
cost = 198

x [*,*]
:    1   2   3   4   5   6   7   8   9   10  11  12    :=
1    0   1   0   0   0   0   0   0   0   0   0   0
2    0   0   0   0   0   0   1   0   0   0   0   0
3    0   0   0   0   0   0   0   0   0   0   0   1
4    0   0   0   0   0   0   0   0   0   1   0   0
5    0   0   1   0   0   0   0   0   0   0   0   0
6    0   0   0   0   0   0   0   0   0   0   1   0
7    0   0   0   1   0   0   0   0   0   0   0   0
8    0   0   0   0   0   1   0   0   0   0   0   0
9    0   0   0   0   1   0   0   0   0   0   0   0
10   1   0   0   0   0   0   0   0   0   0   0   0
11   0   0   0   0   0   0   0   1   0   0   0   0
12   0   0   0   0   0   0   0   0   1   0   0   0
;

ampl:
```

1) The objective value is coming to be: **198**
2) The facility at each location should be:

LOCATION ‘i’	FACILITY ASSIGNED
1	10
2	1
3	5

4	7
5	9
6	8
7	2
8	11
9	12
10	4
11	6
12	3

Q5) [R]

We have changed the integer variables in our model to continuous variables. i.e, we have removed the condition of it to be integer. The code files(ex1b.mod,ex1b.dat,ex1b.run) are tagged in the zipped file.

Console

AMPL

```
ampl: include ex1b.run;
Gurobi 8.0.0: optimal solution; objective 198
4 simplex iterations
cost = 198

x [*,*]
:     1   2   3   4   5   6   7   8   9   10  11  12    :=
1     0   1   0   0   0   0   0   0   0   0   0   0
2     0   0   0   0   0   0   1   0   0   0   0   0
3     0   0   0   0   0   0   0   0   0   0   0   1
4     0   0   0   0   0   0   0   0   0   1   0   0
5     0   0   1   0   0   0   0   0   0   0   0   0
6     0   0   0   0   0   0   0   0   0   0   1   0
7     0   0   0   1   0   0   0   0   0   0   0   0
8     0   0   0   0   0   1   0   0   0   0   0   0
9     0   0   0   0   1   0   0   0   0   0   0   0
10    1   0   0   0   0   0   0   0   0   0   0   0
11    0   0   0   0   0   0   0   1   0   0   0   0
12    0   0   0   0   0   0   0   0   1   0   0   0
;

ampl:
```

The objective value is coming to be: **198**

The facility at each location should be:

LOCATION ‘i’	FACILITY ASSIGNED
1	10
2	1
3	5
4	7
5	9
6	8
7	2
8	11
9	12

10	4
11	6
12	3

The objective value and solutions are coming as that of the above problem when we took it as integer.

Q6) [R] yes,

We will see why it is happening so by taking a very small example , where n=2 and the cost matrix

C=

1	2
2	5

Now , if we solve the question by taking restrictions that x_{ij} can only be integer, then the answer will come come to be

X=

0	1
1	0

and the optimal cost will come to be 4 but if we exclude the restriction of x_{ij} to be ingeter and allow x_{ij} to take any value between 0 1nd 1 i.e, $0 \le x_{ij} \le 1$, then also the objective value will be 4.

For example if we take any other values, then the objective will always be greter than 4.

There can be examplex where we can get fractional values of x . For example if C=

1	1
1	1

then one of the answers can be

x_{11}=0.5

x_{12}=0.5

x_{21}=0.5

x_{22}=0.5

that well give the optimal value, but this type of case is not happening in our question , so in order to get the optimality it will automatically assign locations so that cost would come out to be minimum.

Because if it give floating values to x b/w 0 and 1 then that value will always be greater then or equal to the value we are getting by taking it as integers.

Q7) [R]

The solution(values of x) will always come to be integers (0 or 1) but objective value can get fractional. We have tried giving different fractional values to cost and solved it using AMPL and the code files are attached in the zipped file(ex1c.mod,ex1c.dat,ex1c.run)

Console

AMPL

```
ampl: include ex1c.run;
Gurobi 8.0.0: optimal solution; objective 191.4
17 simplex iterations
cost = 191.4

x [*,*]
:    1   2   3   4   5   6   7   8   9   10  11  12    :=
1    0   0   0   0   0   0   0   0   0   1   0   0
2    0   0   0   0   0   0   1   0   0   0   0   0
3    0   0   0   0   0   0   0   0   0   0   0   1
4    0   0   0   0   1   0   0   0   0   0   0   0
5    0   1   0   0   0   0   0   0   0   0   0   0
6    0   0   0   0   0   0   0   0   0   0   1   0
7    0   0   0   1   0   0   0   0   0   0   0   0
8    0   0   0   0   0   1   0   0   0   0   0   0
9    0   0   1   0   0   0   0   0   0   0   0   0
10   1   0   0   0   0   0   0   0   0   0   0   0
11   0   0   0   0   0   0   0   1   0   0   0   0
12   0   0   0   0   0   0   0   0   1   0   0   0
;

ampl:
```

Now The objective value is coming to be 191.4, but the solutions as we can see in the snapshot clearly are o and 1 only.

Q8) [R]

The changes have made in the mod file and the code files are attached in the zipped file(ex1d.mod,ex1d.run,ex1d.dat).

Console

AMPL

```
ampl: include ex1d.run;
Gurobi 8.0.0: optimal solution; objective 198
17 simplex iterations
cost = 198

x [*,*]
:    1   2   3   4   5   6   7   8   9   10  11  12    :=
1    0   1   0   0   0   0   0   0   0   0   0   0
2    0   0   0   0   0   0   1   0   0   0   0   0
3    0   0   0   0   0   0   0   0   0   0   0   1
4    0   0   0   0   0   0   0   0   0   1   0   0
5    0   0   1   0   0   0   0   0   0   0   0   0
6    0   0   0   0   0   0   0   0   0   0   1   0
7    0   0   0   1   0   0   0   0   0   0   0   0
8    0   0   0   0   0   1   0   0   0   0   0   0
9    0   0   0   0   1   0   0   0   0   0   0   0
10   1   0   0   0   0   0   0   0   0   0   0   0
11   0   0   0   0   0   0   0   1   0   0   0   0
12   0   0   0   0   0   0   0   0   1   0   0   0
;

ampl:
```

The changes that we have made are we have introduced 3 more constraints according to the question.

con3:x[1,3]=0;#i.e, facility 1 cannot be assigned to location 3.
con4:x[12,12]=0;#i.e, facility 12 cannot be assigned to location 12.
con5:x[7,7]=0;#i.e, facility 7 cannot be assigned to location 7.

The objective value of the cost is coming out to be:10
The solution is coming out to be:

(the following facilities have been assigned to the following locations)

LOCATION ‘i’	FACILITY ASSIGNED
1	10
2	1
3	5
4	7
5	9
6	8
7	2
8	11
9	12
10	4
11	6
12	3

In terms of decision variables the answer is

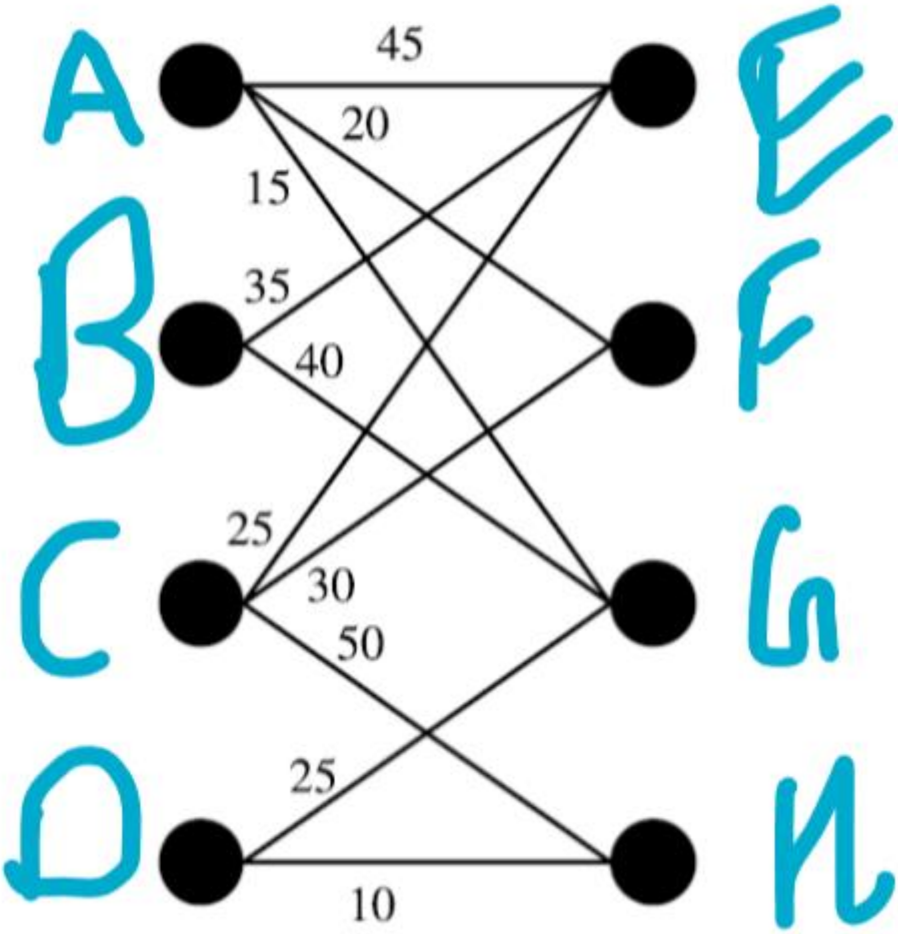
```
:     1   2   3   4   5   6   7   8   9  10  11  12     :=

1     0   1   0   0   0   0   0   0   0   0   0   0
2     0   0   0   0   0   0   1   0   0   0   0   0
3     0   0   0   0   0   0   0   0   0   0   0   1
4     0   0   0   0   0   0   0   0   0   1   0   0
5     0   0   1   0   0   0   0   0   0   0   0   0
6     0   0   0   0   0   0   0   0   0   0   1   0
7     0   0   0   1   0   0   0   0   0   0   0   0
8     0   0   0   0   0   1   0   0   0   0   0   0
9     0   0   0   0   1   0   0   0   0   0   0   0
10    1   0   0   0   0   0   0   0   0   0   0   0
11    0   0   0   0   0   0   0   1   0   0   0   0
12    0   0   0   0   0   0   0   0   1   0   0   0
    ;
```

The above table is showing all the values of x_{ij}.

Exercise 2):

Q1) [R]

As, the graph is not labelled in our question ,let us first label our graph as follows.



(a) Graph 1

then to our intuition ,S can be **S={A,B,C,D}** and the corresponding **V\S = {E,F,G,H}** , then sum of all the weights of the edges from vertices of S to V\S will be **295**. The given answer is correct as every edge of the graph is in the edges from S to V\S. This is happening as there is no edge in GRAPH

1 that that has vertices only from S or only from V\S. Hence, the answer is correct.

Q2) [R]

We have to formulate Integer programming for max cut.

Let G=(V,E) be a given graph.We have to partition the vertices into two parts S and V\S such that weight W_s of the cut is maximum.

<u>Defining the parameters:-</u>

1) Let us define the first parameter N, where N is a natural number that will tell us the number of vertices in the graph.
2) Set S={1,2,3,……….,N} which has n elements which will help us defining our decision variables.
3) defining a matrix $EDGES_{S*S}$ that will tell us the weight of the edges. For example , $EDGES_{23}$ will be the weight of the edge from vertex 2 to 3, and , if there is no vertex from edge 2 to 3 , then value of the weight will be zero.

<u>Defining the decision variables</u> :-

1) let V_i such that i€S, be the decision variables that will tell us whether the given vertex is in S(S is the partition as defined in the definition of the max-cut) or not that is ,

   V_i={1, if the i_{th} vertex is in S

   0, if the i_{th} vertex is not in S
2) let V_s_i such that i€S, be the decision variables that will tell us whether the given vertex is in V\S(V\S is the other

partition as defined in the definition of the max-cut) or not, that is ,

V_s_i={1, if the i_{th} vertex is in V\S

0, if the i_{th} vertex is not in V\S

Then our objective function is :

MAXIMIZE dist=$\sum_{i,j=1}^{N} \boldsymbol{EDGES}\,[\boldsymbol{i},\boldsymbol{j}] * \boldsymbol{V}[\boldsymbol{i}] * \boldsymbol{V\_s}[\boldsymbol{j}]$

, where *EDGES[i,j]*=$EDGES_{ij}$, *V[i]*=V_i , *V_s[j]*=$V\_s_{j.}$

But, the above equation is non-linear. In order to make it linear, we will introduce some new decision variables:

Let X_{ij} be a variable that is **binary** for all i,j€S(we will compel this variable to be V_i*V_s_j by introducing some addition constrains)

<u>Now, defining the objective function</u>:-

MAXIMIZE dist=$\sum_{i,j=1}^{N} \boldsymbol{EDGES}\,[\boldsymbol{i},\boldsymbol{j}] * \boldsymbol{X}[\boldsymbol{i},\boldsymbol{j}]$

, where *EDGES[i,j]*=$EDGES_{ij}$, *X[i,j]*=X_{ij} .

<u>Defining the constraints</u>:-

1) (As a vertex can only be either in S or V\S)

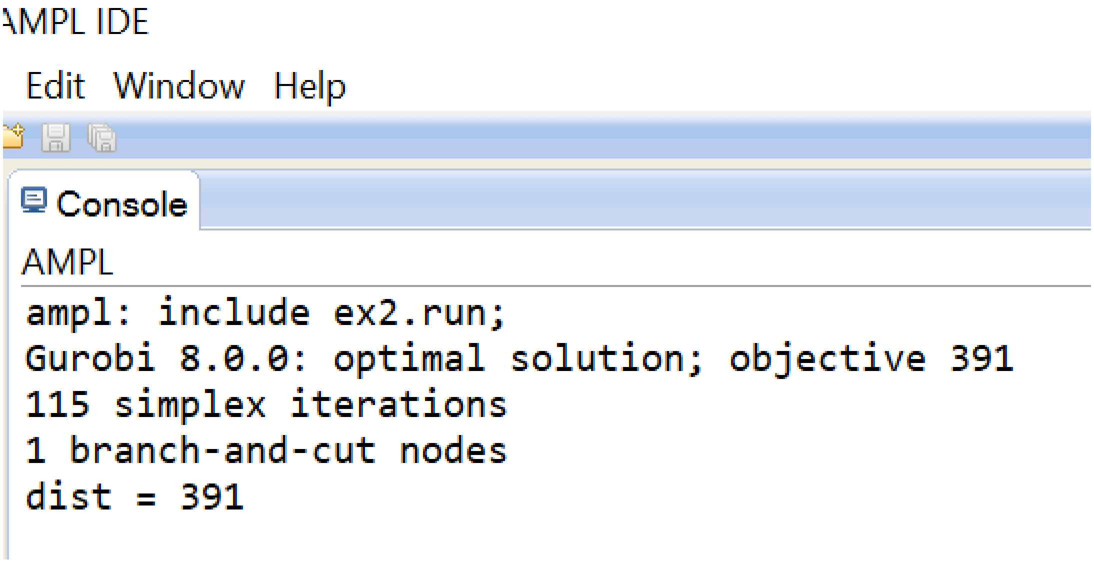
V_i+V_s_i=1, for all i€S.

(Now we will define the linear decision variables that will help us to maintain X_{ij}=V_i*V_s_j)

2) $X_{ij} \leq V_i$ for all i,j€S.
3) $X_{ij} \leq V\_s_j$ for all i,j€S.
4) $X_{ij} \geq V_i + V\_s_j - 1$.

Q4) [R]

We have solve the max-cut of the graph 2 using the way we defined above. The problem has been solved using AMPL and the code file(ex2.mod,ex2.run,ex2.dat) are attached in the zipped files.


AMPL IDE
Edit Window Help
Console
AMPL
ampl: include ex2.run;
Gurobi 8.0.0: optimal solution; objective 391
115 simplex iterations
1 branch-and-cut nodes
dist = 391

```
:    A    B    C    D    E    F    G    H    I      :=
A    0    0    0    0    0    0    0    0    0
B    1    0    0    0    1    1    0    1    1
C    1    0    0    0    1    1    0    1    1
D    1    0    0    0    1    1    0    1    1
E    0    0    0    0    0    0    0    0    0
F    0    0    0    0    0    0    0    0    0
G    1    0    0    0    1    1    0    1    1
H    0    0    0    0    0    0    0    0    0
I    0    0    0    0    0    0    0    0    0
;

v [*] :=
A  0
B  1
C  1
D  1
E  0
F  0
G  1
H  0
I  0
;

v_s [*] :=
A  1
B  0
C  0
D  0
E  1
F  1
G  0
H  1
I  1
;
```

The optimal objective value is coming to be: 391

Solution is :-

Partition 1 will be **S={B,C,D,G}** and **V\S = {A,E,F,H,I}.**

(Here S and V\S are the notations as per defined in the definition of the question given)

Exercise 3):

Q1) [R]

We have to formulate the integer liner programming of the above question:

<u>Defining the parameters:</u>

1) N, where N belongs to the natural numbers.., this will help us making the set of locations and factories.
2) Defining a set LOCATION={1,2,3,.....N}, each element corresponds to each location.
3) Defining a set FACTORY={1,2,3,.....N}, each element corresponds to each Factory.
4) Defining a matrix COSTM=[c_{ij}] , where i,j€LOCATION. So , basically this matrix is defined on LOCATION*LOCATION. This matrix will tell Unit cost of transportation from a location to another location.
5) Defining a matrix TRIPSM=[t_{ij}], where i,j€FACTORY. This matrix will tell us Quantities that must be transported from a factory to another factory every week.

<u>Defining the Decision variables:</u>

1) X_{ij} , basically these variables are defines on a matrix of FACTORY*LOCATION, X_{ij} is equal to 1 if i^{th} factory is assigned to j^{th} location and 0 else.

2) We define some other **binary** decision variables that will help us making the problem linear. Let Y_{ijkr}, is such that i,k€FACTORY and j,r€LOCATION.

<u>Defining the objective function:</u>

MINIMIZE cost:$\sum_{i \text{ in FACTORY}} \sum_{j \text{ in LOCATION}} \sum_{k \text{ in FACTORY}} \sum_{r \text{ in LOCATION}} TRIPSM_{i,k} * COSTM_{j,r} * Y_{i,j,k,r}$

<u>Now , defining the constraints:</u>

1) As we know that one factory can go only one location: $\sum_{j=1}^{12} Xij$ = 1 , for all i€S.

2)As we know that one location can have only one factory:

$\sum_{i=1}^{12} Xij$ = 1 , for all j€S.

(Our next constraints will be to make $Y_{ijkr}=X_{ij}*X_{kr}$)

3) $Y_{ijkr} \leq X_{ij}$ for all i,k€FACTORY, j,r€LOCATION.
4) $Y_{ijkr} \geq X_{kr} + X_{ij} - 1$
5) $Y_{ijkr}=Y_{krij}$

(In the above question we have introduced the aove variable y so that we can linearize the above model , if we donot linearize the above model then we can write Y_{ijkr} as $X_{ij*}X_{kr}$ and remove constraints 4,5,6).

Q2) [R]

If there are n factories and n locations, then we have to tell the no. of constraints and no. of variables are as follows.

(Y corresponds to n^4 variables and X corresponds to n^2 variables.)

NO. OF VARIABLES:= n^4 + n^2.

We are calculating the no. of constraints using the model described above as if there were n locations and factories instead of n.

1 corresponds to n constraints.

2 corresponds to n constraints.

3 corresponds to n^4 constraints.

4 corresponds to n^4 constraints.

5 corresponds to n^4 constraints.

NO. OF CONTRAINTS: $3*n^4 + 2*n$.

Q3) [R]

We have solved the following question using ampl but it is taking too much time because of too many variables and constraints , so the command is interrupted and the results are as shown (code file are ex3.mod,ex3.run,ex3.dat) following result:-

```
18i190002@tunga:~/practiceiitb/ampl$ ampl ex3.run
Gurobi 8.0.0: timing 1
^C
<BREAK> (gurobi)

<BREAK> (gurobi)

Times (seconds):
Input =  0.06
Solve =  600.204 (summed over threads)
Output = 0.032
Elapsed = 147
Gurobi 8.0.0: interrupted with a feasible solution; objective 40192
12888 simplex iterations
165 branch-and-cut nodes
No basis.
No dual variables returned.
cost = 40192

x [*,*]
:    1   2   3   4   5   6   7   8   9   10  11  12    :=
1    0   0   0   0   0   1   0   0   0   0   0   0
2    0   1   0   0   0   0   0   0   0   0   0   0
3    0   0   0   0   0   0   0   0   1   0   0   0
4    0   0   0   0   0   0   0   0   0   0   1   0
5    0   0   0   1   0   0   0   0   0   0   0   0
6    0   0   1   0   0   0   0   0   0   0   0   0
7    0   0   0   0   0   0   0   1   0   0   0   0
8    0   0   0   0   0   0   1   0   0   0   0   0
9    1   0   0   0   0   0   0   0   0   0   0   0
10   0   0   0   0   0   0   0   0   0   1   0   0
11   0   0   0   0   0   0   0   0   0   0   0   1
12   0   0   0   0   1   0   0   0   0   0   0   0
;

18i190002@tunga:~/practiceiitb/ampl$
```

cost coming is 40192

Location1=9

Location2=2

Location3=6

Location4=5

Location5=12

Location6=1

Location7=8

Location8=7

Location9=3

Location10=10

Location11=4

Location12=11

time taken is **600.204** sec