**PROGRAM FOR PRIM’S ALGORITHM:**

#include<iostream>

#include<stdio.h>

#include<iomanip>

using namespace std;

class Prims

{

private:

int visited[25];

int A[25][25];

int houses,min,cost;

public:

Prims()

{

houses=0;

min=99;

cost=0;

}

void accept();

void print\_dist();

};

void Prims :: accept()

{

cout<<"Enter the no. of houses per row(Here,i.e. equal to no. of rows)";

cin>>houses;

for(int i=0;i<houses;i++)

{

visited[i]=0;

}

cout<<"\nEnter the Adjecency Matrix For Distances:\n";

for(int i=0;i<houses;i++)

{

for(int j=0;j<houses;j++)

{

cin>>A[i][j];

}

cout<<"\nEnter the next row\n";

}

cout<<"\nYour Entered Adjecency Matrix is:\n";

for(int i=0;i<houses;i++)

{

for(int j=0;j<houses;j++)

{

cout<<setw(10)<<A[i][j];

}

cout<<endl;

}

}

void Prims :: print\_dist()

{

int k,l;

for(int i=0;i<houses;i++)

{

for(int j=0;j<houses;j++)

{

if(A[i][j]<min && A[i][j]!=0)

{

min = A[i][j];

k=i;

l=j;

}

}

}

cout<<"\nMinimum at("<<k<<","<<l<<") is :"<<min;

cost = min;

visited[k]=1;

visited[l]=1;

for(int n=0;n<houses-2;n++)

{

min =99;

for(int i=0;i<houses;i++)

{

if(visited[i]==1)

{

for(int j=0;j<houses;j++)

{

if(visited[j]==0)

{

if(A[i][j] < min && A[i][j]!=0)

{

min = A[i][j];

k=i;

l=j;

}

}

}

}

}

visited[k]=1;

visited[l]=1;

cost = cost + min;

}

cout<<"\nThe cost is:"<<cost;

}

int main()

{

Prims P1;

P1.accept();

P1.print\_dist();

return 0;

}

**OUTPUT**:

Enter the no. of houses per row(Here,i.e. equal to no. of rows)6

Enter the Adjecency Matrix For Distances:

0

6

1

5

0

0

Enter the next row

6

0

5

0

3

0

Enter the next row

1

5

0

5

6

4

Enter the next row

5

0

5

0

0

2

Enter the next row

0

3

6

0

0

6

Enter the next row

0

0

4

2

6

0

Enter the next row

Your Entered Adjecency Matrix is:

0 6 1 5 0 0

6 0 5 0 3 0

1 5 0 5 6 4

5 0 5 0 0 2

0 3 6 0 0 6

0 0 4 2 6 0

Minimum at(0,2) is :1

The cost is:15