**ASSIGNMENT**

**Prims Algorithm**

**QUESTION 1**

#include <iostream>

#include <conio.h>

using namespace std;

struct node

{

int f, t, co;

}p[6];

int c = 0, t1 = 0, t = 0;

void prims(int \*a, int b[][4], int i, int j)

{

a[i] = 1;

while (c < 3)

{

int min = 999;

for (int i = 0; i < 4; i++)

{

if (a[i] == 1)

{

for (int j = 0; j < 4; )

{

if (b[i][j] >= min || b[i][j] == 0)

{

j++;

}

else if (b[i][j] < min)

{

min = b[i][j];

t = i;

t1 = j;

}

}

}

}

a[t1] = 1;

p[c].f = t;

p[c].t = t1;

p[c].co = min;

c++;

b[t][t1] = b[t1][t]=1000;

}

for (int k = 0; k < 3; k++)

{

cout<<"source node:"<<p[k].f<<endl;

cout<<"destination node:"<<p[k].t<<endl;

cout<<"weight of node"<<p[k].co<<endl;

}

}

int main()

{

int a[4];

for (int i = 0; i < 4; i++)

{

a[i] = 0;

}

int b[4][4];

for (int i = 0; i < 4; i++)

{

cout<<"enter values for "<<(i+1)<<" row"<<endl;

for (int j = 0; j < 4; j++)

{

cin>>b[i][j];

}

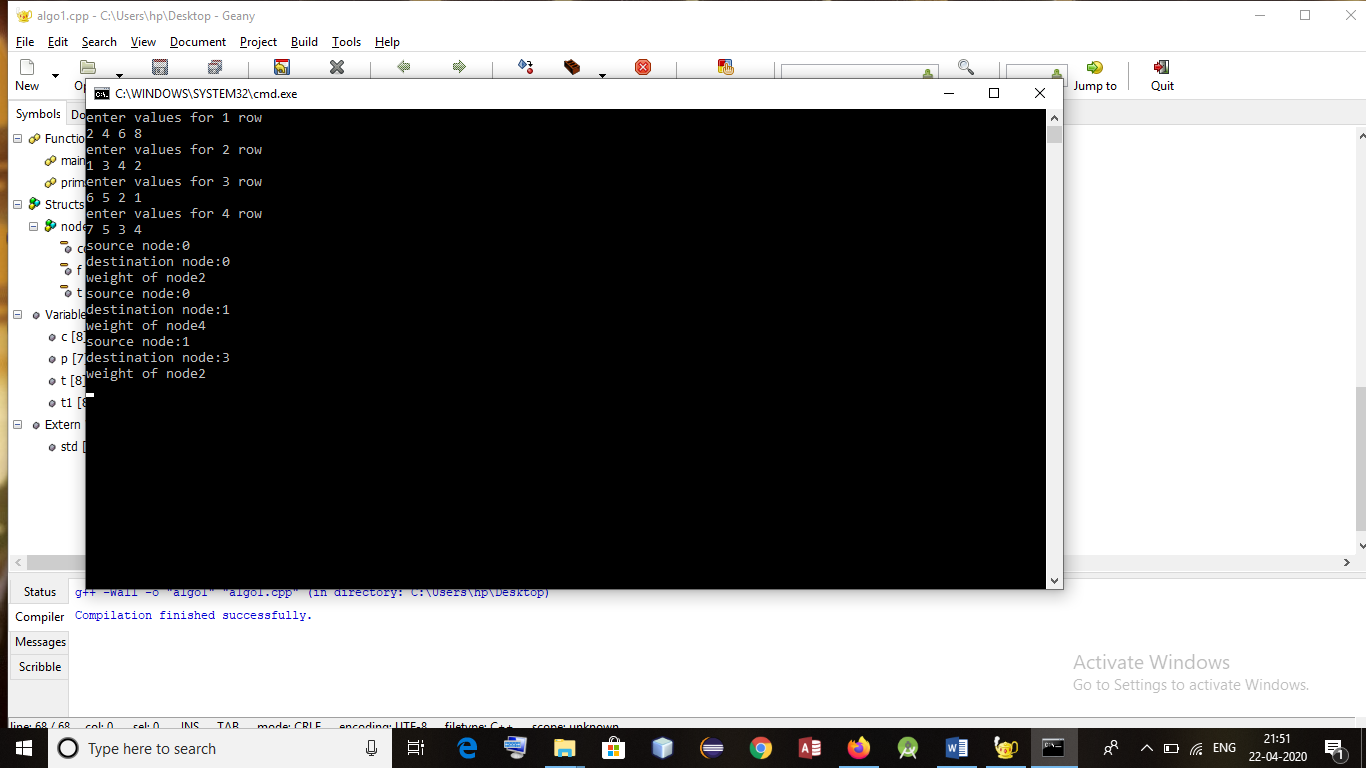
}

prims(a, b, 0, 0);

getch();

}

**Output**



**Kruskals Algorithm**

**QUESTION 2**

#include<iostream>

#include<conio.h>

#include<stdlib.h>

using namespace std;

int co[10][10],i,j,k,n,m,c,visit,visited[10],l,v,count,count1,vst,p;

main()

{

int d1,d2;

cout<<"enter no of vertices";

cin >> n;

cout <<"enter no of edges";

cin >>m;

cout <<"Edge ";

for(k=1;k<=m;k++)

{

cin >>i >>j >>c;

co[i][j]=c;

co[j][i]=c;

}

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

if(co[i][j]==0)

co[i][j]=31999;

visit=1;

while(visit<n)

{

v=31999;

for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

if(co[i][j]!=31999 && co[i][j]<v && co[i][j]!=-1 )

{

int count =0;

for(p=1;p<=n;p++)

{

if(visited[p]==i || visited[p]==j)

count++;

}

if(count >= 2)

{

for(p=1;p<=n;p++)

if(co[i][p]!=31999 && p!=j)

d1=p;

for(p=1;p<=n;p++)

if(co[j][p]!=31999 && p!=i)

d2=p;

if(co[d1][d2]==-1)

continue;

}

l=i;

k=j;

v=co[i][j];

}

cout <<"\tedge from " <<l <<"-->\t"<<k;

co[l][k]=-1;

co[k][l]=-1;

visit++;

int count=0;

count1=0;

for(i=1;i<=n;i++)

{

if(visited[i]==l)

count++;

if(visited[i]==k)

count1++;

}

if(count==0)

visited[++vst]=l;

if(count1==0)

visited[++vst]=k;

}

}

**Output**

