#include<iostream>

using namespace std;

class edge

{

public:

int u,v,wt;

};

int check(edge e, int path[50])

{

int u=e.u, v=e.v;

while(path[u]>0)

u=path[u];

while(path[v]>0)

v=path[v];

if(u!=v)

{

path[u]=v;

return 1;

}

return 0;

}

int main()

{

int n,m,mincost,i,j;

cout<<" \n enter the no. of vertices in the graph";

cin>>n;

cout<<" \n enter the no. of edges in the graph";

cin>>m;

int path[50];

edge e[50];

edge t;

for(i=0;i<m;i++)

{

cout<<"enter the two vertices and the edge of the graph\n"<<i+1<<endl;

cout<<"first vertex\n";

cin>>e[i].u;

cout<<"second vertex\n";

cin>>e[i].v;

cout<<"enter weight\n";

cin>>e[i].wt;

cout<<endl;

}

for(i=0;i<=m-1;i++)

{

for(j=0;j<m-i-1;j++)

{

if(e[j].wt>e[j+1].wt)

{

t=e[j];

e[j]=e[j+1];

e[j+1]=t;

}

}

};

for(i=0;i<=n;i++)

{

path[i]=0;

}

i=0;

//count the no. of edges selected or discarded in a tree

j=0;

mincost=0;

cout<<endl;

while((i!=n-1)&&(j!=m)){

if(check(e[j],path))

{

mincost=mincost+e[j].wt;

i++;

cout<<"edge("<<e[j].u<<","<<e[j].v<<")"<<"with weight"<<e[j].wt<<"";

cout<<"is selected"<<endl;

}

j++;

}

cout<<endl<<"MINIMUM COST OF TREE IS: "<<mincost<<endl;

return 0;

}

Output:

enter the no. of vertices in the graph5

enter the no. of edges in the graph5

enter the two vertices and the edge of the graph

1

first vertex

1

second vertex

2

enter weight

3

enter the two vertices and the edge of the graph

2

first vertex

1

second vertex

3

enter weight

2

enter the two vertices and the edge of the graph

3

first vertex

1

second vertex

4

enter weight

8

enter the two vertices and the edge of the graph

4

first vertex

1

second vertex

5

enter weight

6

enter the two vertices and the edge of the graph

5

first vertex

2

second vertex

3

enter weight

1

edge(2,3)with weight1is selected

edge(1,3)with weight2is selected

edge(1,5)with weight6is selected

edge(1,4)with weight8is selected

MINIMUM COST OF TREE IS: 17