#include<iostream>

using namespace std;

class MST

{

int a[20][20],n,k;

struct gr

{

int v1;

int v2;

int wt;

}g[20];

public:

void accept();

void extract\_edges();

void prims();

};

void MST::accept()

{

int i,j;

cout<<"\nEnter the no. of vertices: ";

cin>>n;

cout<<"Enter adjacency matrix:\n";

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

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

cin>>a[i][j];

}

void MST::extract\_edges()

{

int i,j;

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

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

if(a[i][j]!=0 )

{

g[k].v1=i;

g[k].v2=j;

g[k++].wt=a[i][j];

}

cout<<"Edges in the graph are:\n\tSource\tSink\tWeight\n";

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

cout<<"\t"<<char(g[i].v1+65)<<"\t"<<char(g[i].v2+65)<<"\t"<<g[i].wt<<"\n";

}

void MST::prims()

{

int i,j,min\_edge,visited[20]={0},sum=0,min,flag;

visited[0]=1;

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

{

min=1000;

flag=0;

for(j=0;j<k;j++)

{

if((visited[g[j].v1]==0&&visited[g[j].v2]==1)||(visited[g[j].v1]==1&&visited[g[j].v2]==0))

if(g[j].wt<min)

{

min=g[j].wt;

min\_edge=j;

flag=1;

}

}

if(flag)

{

cout<<"\nEdge included "<<char(g[min\_edge].v1+65)<<"--"<<char(g[min\_edge].v2+65)<<" with weight "<<g[min\_edge].wt;\

visited[g[min\_edge].v1]=1;

visited[g[min\_edge].v2]=1;

sum+=g[min\_edge].wt;

}

}

cout<<"\nTotal cost of MST: "<<sum<<"\n";

}

main()

{

MST m;

m.accept();

m.extract\_edges();

m.prims();

return 0;

}