**ASSIGNMENT NO.**

NAME: SHRUTI DILIP BHUJANGE

CLASS: BE COMP-1 ROLL NO.: 402006

PROGRAM:

#include<iostream>

using namespace std;

int visit[10];

int temp[10];

int no=5;

struct node

{

int mat[10][10];

int red\_cost;

int nodeno;

struct node \*next[10];

int level;

}\*head;

int reduction(node\* n)

{

int min[100];

int red=0;

//row reduction

for(int i=1;i<=no;i++)

{

min[i]=999;

for(int j=1;j<=no;j++)

{

if(n->mat[i][j]<min[i])

{

min[i]=n->mat[i][j];

}

}

}

for(int i=1;i<=no;i++)

{

for(int j=1;j<=no;j++)

{

if(min[i]!=999 and i!=j and n->mat[i][j]!=999)

{

n->mat[i][j]=n->mat[i][j]-min[i];

//cout<<"\t"<<min[i];

}

}

}

//cout<<"\nrow:\n";

for(int i=1;i<=no;i++)

{

if(min[i]!=999)

red=red+min[i];

//cout<<min[i]<<"\t";

}

//column reduction

for(int i=1;i<=no;i++)

{

min[i]=999;

for(int j=1;j<=no;j++)

{

if(n->mat[j][i]<min[i])

{

min[i]=n->mat[j][i];

}

}

}

for(int i=1;i<=no;i++)

{

for(int j=1;j<=no;j++)

{

if(min[i]!=999 and i!=j and n->mat[j][i]!=999)

{

n->mat[j][i]=n->mat[j][i]-min[i];

//cout<<"\t"<<min[i];

//red=red+min[i];

}

}

}

//cout<<"\ncol:\n";

for(int i=1;i<=no;i++)

{

if(min[i]!=999)

red=red+min[i];

//cout<<min[i]<<"\t";

}

return red;

}

void display(node \*n)

{

for(int i=1;i<=no;i++)

{

for(int j=1;j<=no;j++)

{

cout<<n->mat[i][j]<<"\t";

}

cout<<"\n";

}

}

int findNextj()

{

for(int i=1;i<=no;i++)

{

if(temp[i]==0)

{

temp[i]=1;

return i;

}

}

return -1; //if -1 : all nodes visited

}

struct node\* nextNodes(node \*n1)

{

int j=1;

struct node\* curr;

curr=new node();

for(int s=0;s<=no;s++)

{

temp[s]=visit[s];

}

for(int k=1;k<=no-(n1->level);k++)

{

int i=n1->nodeno;

j=findNextj();

if(j==-1)

{

cout<<"\nAll nodes visited";

break;

}

n1->next[k]=new node();

for(int p=1;p<=no;p++)

{

for(int q=1;q<=no;q++)

{

n1->next[k]->mat[p][q]=n1->mat[p][q];

}

}

//n1->next[k]->mat=n1->mat;

curr=n1->next[k];

for(int l=1;l<=no;l++)

{

curr->mat[i][l]=999;

curr->mat[l][j]=999;

}

curr->mat[j][1]=999;

//cout<<"\n\n\n\*\*\*\*\*\*Before reduction\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

//display(n1->next[k]);

curr->red\_cost=reduction(curr)+n1->red\_cost+n1->mat[i][j];

curr->nodeno=j;

n1->next[k]=curr;

}

}

struct node \*findMin(struct node \*c)

{

int min=999;

struct node\* curr;

for(int k=1;k<=no-(c->level);k++)

{

if((c->next[k]->red\_cost)<min)

{

min=(c->next[k])->red\_cost;

curr=c->next[k];

}

}

return curr;

}

int allnv()

{

for(int i=1;i<=no;i++)

{

if(visit[i]==0)

{

return 0;

}

}

return 1;

}

int main()

{

head=new node();

int path[10],h=1;

int cost=0;

cout<<endl;

for(int i=1;i<=no;i++)

{

visit[i]=0;

}

head->mat[1][1]=999; head->mat[1][2]=20; head->mat[1][3]=30; head->mat[1][4]=10; head->mat[1][5]=11;

head->mat[2][1]=15; head->mat[2][2]=999; head->mat[2][3]=16; head->mat[2][4]=4; head->mat[2][5]=2;

head->mat[3][1]=3; head->mat[3][2]=5; head->mat[3][3]=999; head->mat[3][4]=2; head->mat[3][5]=4;

head->mat[4][1]=19; head->mat[4][2]=6; head->mat[4][3]=18; head->mat[4][4]=999; head->mat[4][5]=3;

head->mat[5][1]=16; head->mat[5][2]=4; head->mat[5][3]=7; head->mat[5][4]=16; head->mat[5][5]=999;

display(head);

head->red\_cost=reduction(head);

head->level=1;

path[1]=1;

head->nodeno=1;

visit[head->nodeno]=1;

struct node \*curr;

curr=head;

struct node\* n1;

n1=new node();

while(!allnv())

{

nextNodes(curr);

n1=findMin(curr);

n1->level=curr->level+1;

path[++h]=n1->nodeno;

visit[n1->nodeno]=1;

cout<<"\n";

curr=n1;

}

cost=n1->red\_cost;

cout<<"\nPath:\t";

for(int h=1;h<=no;h++)

{

cout<<path[h]<<"-->";

}

cout<<head->nodeno;

cout<<endl<<"Cost: "<<cost<<"\n";

}