#include<stdio.h>

#include<stdlib.h>

int visited[10],n,cost[10],l = 0;

static int node\_no = 1;

int a[10][10];

int find\_row\_min(int b[][10],int r)

{

int min = 999,j;

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

{

if(b[r][j] < min)

{

min = b[r][j];

}

}

if(min == 999)

{

min = 0;

}

return min;

}

int find\_col\_min(int b[][10],int c)

{

int i,min=999;

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

{

if(b[i][c] < min)

{

min = b[i][c];

}

}

if(min == 999)

{

min = 0;

}

return min;

}

int min1(int temp[][2],int q)

{

int i,j,small=999;

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

{

if(temp[i][1] < small)

{

small = temp[i][1];

j = i;

}

}

return j;

}

int check\_visited(int k)

{

int i;

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

{

if(visited[i] == k)

{

return 1;

}

}

return 0;

}

void display\_path()

{

int i;

printf("\nThe path of tour is\n");

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

{

printf("\t%d",visited[i]+1);

}

printf("\t1");

printf("\nThe minimum cost of tour is %d\n",cost[l-1]);

}

int reduced\_matrix(int b[10][10])

{

int i,j,reduced\_sum = 0,min;

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

{

min = find\_row\_min(b,i);

reduced\_sum+= min;

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

{

if(b[i][j] != 999)

{

b[i][j] -= min;

}

}

}

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

{

min = find\_col\_min(b,i);

reduced\_sum += min;

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

{

if(b[j][i] != 999)

{

b[j][i] -= min;

}

}

}

printf("\nReduced MAtrix for node no;%d is :\n",node\_no++);

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

{

printf("\n");

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

{

printf("\t%d",b[i][j]);

}

}

return reduced\_sum;

}

void tspbb()

{

int temp[10][2],q,p,m[10][10],sum = 0;

int i,j,k;

while(l < n)

{

q = 0;

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

{

p = check\_visited(i);

if(p == 0)

temp[q++][0] = i;

}

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

{

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

{

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

{

m[k][j] = a[k][j];

}

}

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

{

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

{

m[visited[k]][j]=999;

}

}

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

{

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

{

m[j][visited[k]]=999;

}

}

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

{

m[j][temp[i][0]]=999;

}

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

{

if(visited[j+1]!= -1)

{

m[visited[j+1]][0] =999;

}

}

m[temp[i][0]][0] = 999;

sum = reduced\_matrix(m);

temp[i][1] = cost[l - 1] + sum + a[visited[l-1]][temp[i][0]];

printf("\ncost odf node no %d=%d",node\_no-1,temp[i][1]);

}

p = min1(temp,q);

visited[l] = temp[p][0];

cost[l++] = temp[p][1];

printf("\n\nminimum cost city is =%d and its cost is :%d",temp[p][0]+1,temp[p][1]);

}

}

int main()

{

int i,j;

printf("Enter no of cities\n");

scanf("%d",&n);

printf("Enter cost matrix\n");

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

{

printf("Enter elements of row %d ",i+1);

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

{

scanf("%d",&a[i][j]);

}

visited[i] = -1;

}

visited[l] = 0;

cost[l++] = reduced\_matrix(a);

tspbb();

display\_path();

return 0;

}

OUTPUT:



