#include<stdio.h>

//#include<conio.h>

int rowmin(int ar[4][4],int rm[4][4]);

int colmin(int ar[4][4],int rm[4][4]);

int main()

{

int totalr,reducedmatrix[4][4],reducedmatrix1[4][4],i,j,save\_rm[4][4],mab,cost[3],k;

int ar[4][4]={{999,10,15,20}, // array

{5,999,9,10 },

{6,13,999,12},

{8,8,9,999}};

totalr=rowmin(ar,reducedmatrix); //calling row min

printf("\n\n total = %d\n\n",totalr); //printing cost after column reducion in matrix

for(i=0;i<3;i++)

{

for(j=0;j<=3;j++)

{

save\_rm[i][j]=reducedmatrix[i][j]; //copying reduced matrix in saved reduced matrix

}

}

for(k=1;k<=3;k++)

{

mab=save\_rm[0][k]; //value of mab is equal to saved reduced matrix at position 0 , k

reducedmatrix[k][0]=999; // making reduced matrix at position k ,0 as infinity

for(i=0;i<=3;i++)

{

for(j=0;j<=3;j++)

{

if(i==0 || j==k) //if i== 0 and j== value of k then reduced matrix at position i,j is infinity

{

reducedmatrix[i][j]=999; //making reduced matrix at i , j as infinity

}

}

}

cost[k]=rowmin(reducedmatrix,reducedmatrix1); //calculating cost of row minimized matrix

cost[k]=totalr+cost[k]+mab; //calculating total cost of reduced matrix (formula ; cost = total cost of reduced matrix +cost(of previous martrix) + value of mab )

printf("cost = %d \n",cost[k]); // printing cost

}

}

//calculating min in row

int rowmin(int ar[4][4],int reducedmatrix[4][4])

{

int rowreduction=0,totalreduction;

int min[4]={999,999,999,999},i,j;

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

{

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

{

if(min[i]>ar[i][j])

{

min[i]=ar[i][j];

}

}

if(min[i]==999){min[i]=0;}

rowreduction=rowreduction+min[i]; //adding row min

}

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

{

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

{

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

{

ar[i][j]=ar[i][j]-min[i];

}

printf("%d ",ar[i][j]);

}

printf("\n");

}

totalreduction=colmin(ar,reducedmatrix);

totalreduction=totalreduction+rowreduction; //

return totalreduction;

}

//calculating min in column

int colmin(int ar[4][4],int reducedmatrix[4][4])

{

int colreduction=0;

int min[4]={999,999,999,999},i,j;

for( i=0;i<4;i++) //finding min in row matrix

{

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

{

if(min[i]>ar[j][i]) //checking if min is greater than value at that particular instant

{

min[i]=ar[j][i]; //min value found and stored

}

}

if(min[i]==999){min[i]=0;}

colreduction=colreduction+min[i]; //adding total min value in column min matrix

}

for(i=0;i<4;i++) // for loop for subtracting min value from each column value

{

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

{ if(ar[j][i]!=999) //checking if column value is

{

ar[j][i]=ar[j][i]-min[i]; //subtracting value at array i,j eith min value in a column

}

}

}

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

{

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

{

reducedmatrix[i][j]=ar[i][j];

printf("%d ",ar[i][j]);

}

printf("\n");

}

return colreduction; //returning value of column reduction

}