Aim:

To write a program to find the minimum cost using north west corner rule,least cost method and vogels approximation method.

Program:

```
#Python 3.6:
def vogelhelp(l):
   new=[]
   for i in range(len(l)-1):
      sor=[[l[i][j],j] for j in range(len(l[i])-1) if l[i][j] is not None]
      sor=sorted(sor,key=lambda x:x[0])
      if len(sor)>=2:
         obj1=sor[0];obj2=sor[1]
         new.append([abs(obj1[0]-obj2[0]),i,obj1[1]])
   l1=[i for i in zip(*l)][1:]
   for i in range(len(l1)-1):
      sor=[[l1[i][j],j] for j in range(len(l1[i])-1) if l1[i][j] is not None]
      sor=sorted(sor,key=lambda x:x[0])
      if len(sor)>=2:
         obj1=sor[0];obj2=sor[1]
         new.append([abs(obj1[0]-obj2[0]),obj1[1],i+1])
   try:
      maxi=max(new,key=lambda x:x[0])
      return [maxi[1],maxi[2],l[maxi[1]][maxi[2]]]
   except:
      for i in range(n):
         if l[i][-1]!=0 and l[i][-1] is not None:
            for j in range(1,m+1):
               if l[i][j]!=0 and l[i][j] is not None:
                  objec=[i,j,l[i][j]]
      return objec
def minnes(l):
    minimum = 10**18
    i_min, j_min = 0, 0
    for i, a in enumerate(l):
        for j, b in enumerate(a):
            if b and b < minimum and b is not None:
                i_min, j_min, minimum = i, j, b
    return [i_min,j_min,minimum]
n,m=map(int,input().split())
l=[[None for j in range(m+2)] for i in range(n+1)]
l1=[[None for j in range(m+2)] for i in range(n+1)]
l2=[[None for j in range(m+2)] for i in range(n+1)]
for i in range(n+1):
    s=input().split()
    for j in range(m+1):
        l[i][j+1]=int(s[j])
        l1[i][j+1]=int(s[j])
        l2[i][j+1]=int(s[j])
def northwest():
    print("North west corner rule")
    l5=[];ans=0;
    for i in range(1,m+1):
        j=0;
        while(l[n][i]!=0):
            if l[j][-1]==0:
                j+=1;
            else:
                if l[n][i]-l[j][-1]<0:
                    l[j][-1]-=l[n][i]
                    l5.append([l[j][i],l[n][i]])
                    l[n][i]=0
                else:
                    c=l[n][i]
                    l[n][i]-=l[j][-1]
                    l5.append([l[j][i],l[j][-1]])
                    l[j][-1]=0
                j+=1
    for i in l5:
        ans+=(i[0]*i[1])
        print(*i)
    print("Total Cost:",ans)
northwest()
def leastcost():
    print("Least cost method")
    l5=[];ans=0
    while(len(set(l1[n]))!=2):
        mini=minnes(l1[:-1])
        row,col=mini[0],mini[1];obj=mini[2]
        l1[row][col]=None
        if l1[row][-1]-l1[n][col]>=0:
            l1[row][-1]-=l1[n][col]
            obj1=(l1[n][col])
            for i in range(n+1):
                l1[i][col]=None
        else:
            l1[n][col]-=l1[row][-1]
            obj1=l1[row][-1]
            for j in range(1,m+2):
                l1[row][j]=None
        l5.append([obj,obj1])
    for i in l5:
        ans+=(i[0]*i[1])
        print(*i)
    print("Total Cost:",ans)
leastcost()
def vogel():
    print("Vogels approximation method")
    l5=[];ans=0
    while(len(set(l2[n]))!=2):
        mini=vogelhelp(l2)
        row,col=mini[0],mini[1];obj=mini[2]
        l2[row][col]=None
        if l2[row][-1]-l2[n][col]>=0:
            l2[row][-1]-=l2[n][col]
            obj1=(l2[n][col])
            for i in range(n+1):
                l2[i][col]=None
        else:
            l2[n][col]-=l2[row][-1]
            obj1=l2[row][-1]
            for j in range(1,m+2):
                l2[row][j]=None
        l5.append([obj,obj1])
    for i in l5:
        ans+=(i[0]*i[1])
        print(*i)
    print("Total Cost:",ans)
vogel()
#Tester data 1
##5 5
##20 30 50 15 5 100
##10 20 35 25 30 150
##40 10 15 20 5 100
##2 33 20 40 45 200
```

##11 23 25 41 37 175

##175 100 125 160 165 725

#Tester data 2

##3 3

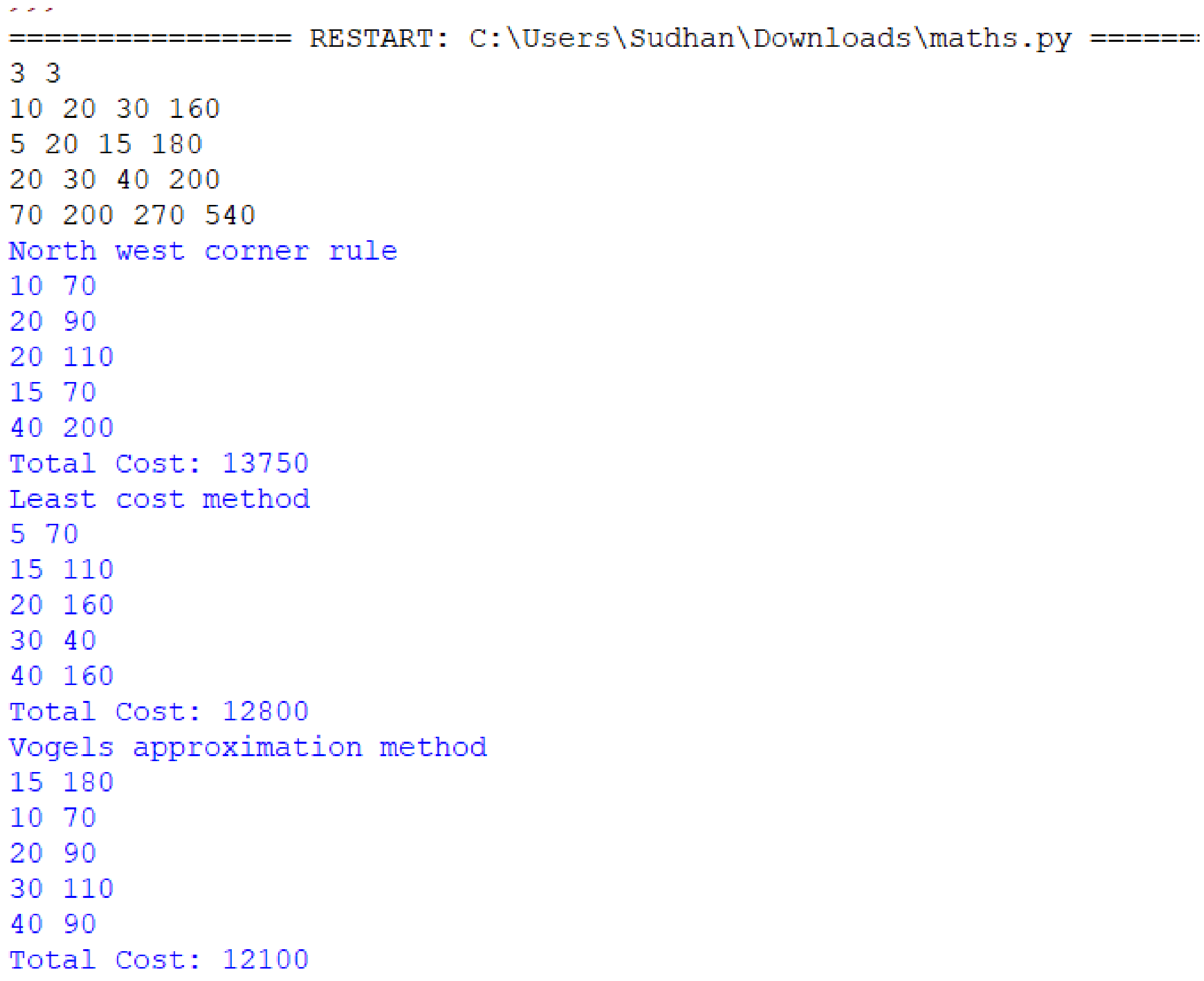
##10 20 30 160

##5 20 15 180

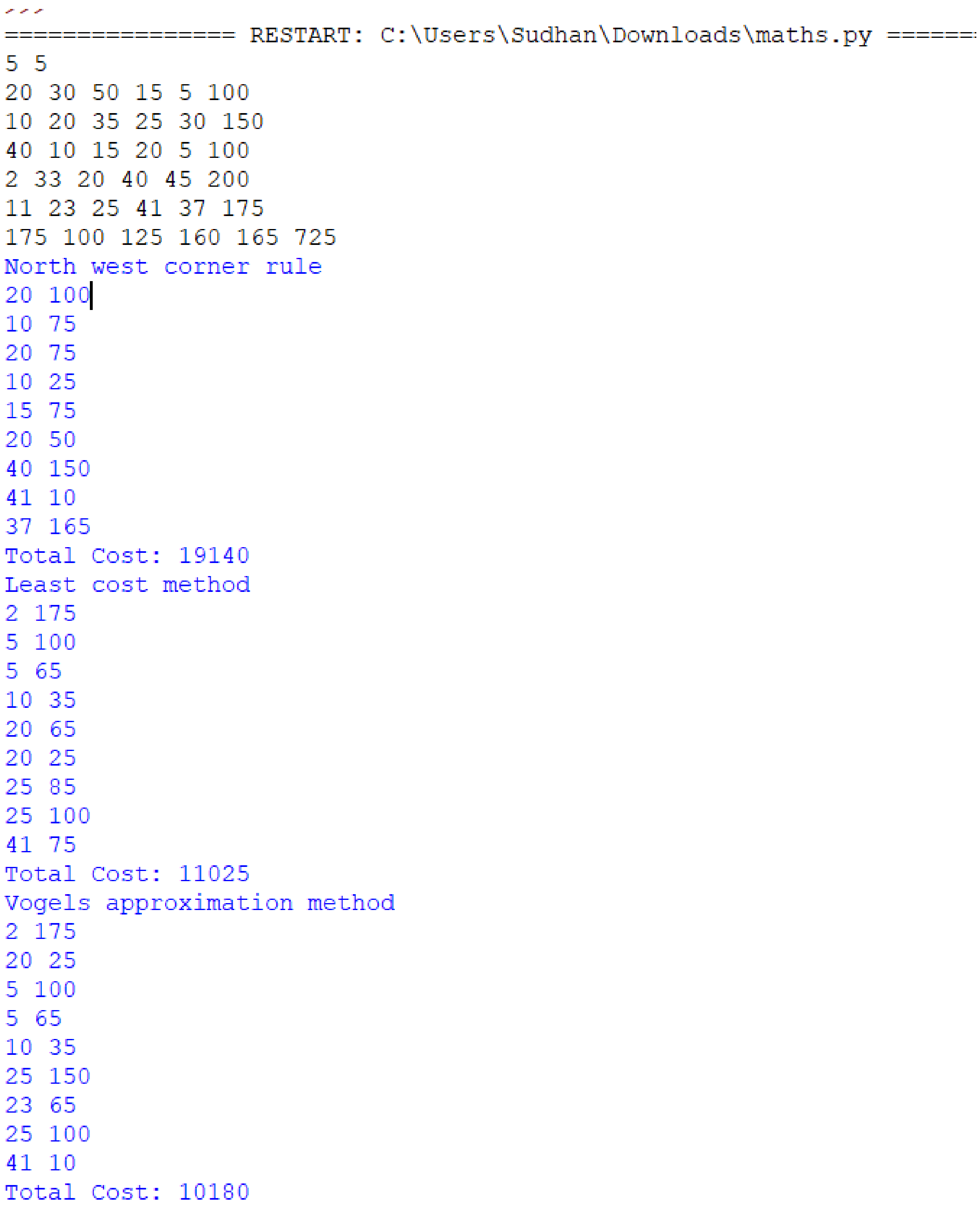
##20 30 40 200

##70 200 270 540

Output:



```
================ RESTART: C:\Users\Sudhan\Downloads\maths.py ======
3 3
10 20 30 160
5 20 15 180
20 30 40 200
70 200 270 540
North west corner rule
10 70
20 90
20 110
15 70
40 200
Total Cost: 13750
Least cost method
5 70
15 110
20 160
30 40
40 160
Total Cost: 12800
Vogels approximation method
15 180
10 70
20 90
30 110
40 90
Total Cost: 12100
```



```
================ RESTART: C:\Users\Sudhan\Downloads\maths.py ======
5 5
20 30 50 15 5 100
10 20 35 25 30 150
40 10 15 20 5 100
2 33 20 40 45 200
11 23 25 41 37 175
175 100 125 160 165 725
North west corner rule
20 100
10 75
20 75
10 25
15 75
20 50
40 150
41 10
37 165
Total Cost: 19140
Least cost method
2 175
5 100
5 65
10 35
20 65
20 25
25 85
25 100
41 75
Total Cost: 11025
Vogels approximation method
2 175
20 25
5 100
5 65
10 35
25 150
23 65
25 100
41 10
Total Cost: 10180
```

Result:

Thus the program has been executed successfully to find the minimum cost using the three methods namely north west corner rule,least cost method and vogels approximation method.