

## Q1

```
timing = [0,12]
#creating the array
arr = [
    ['silent disco', 12, 100000],
    ['bonfire', 12, 500000],
    ['street play', 2, 60000],
    ['dancing competition', 8, 75000],
    ['short film screening', 10, 45000],
    ['rangoli', 4, 50000],
    ['scavenger hunt', 5, 200000],
    ['face art', 4, 40000],
    ['solo music', 7, 95000],
    ['group music', 7, 150000],
    ['mime', 11, 120000],
    ['poetry recitation', 11, 300000]
]
#defining the function
def printJobScheduling(arr, t):
    #length of array
    n=len(arr)
    #sorting all jobs according to decreasing order of penalty
    for i in range(n):
        for j in range(n - 1 - i):
            if arr[j][2] < arr[j + 1][2]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    result = [False] * t #to keep track of time
    job = ['-1'] * t
    for i in range(len(arr)):
        # Find a free slot for this job
        # (Note that we start from the
        # last possible slot)
        for j in range(min(t - 1, arr[i][1] - 1), -1, -1):
            # Free slot found
            if result[j] is False:
                result[j] = True
                job[j] = arr[i][0]
                break
    # print the sequence
    print(job)
    print("Following is maximum profit sequence of jobs")
    printJobScheduling(arr, 12)
```

## answer:

```
['face art', 'street play', 'short film screening',
'rangoli', 'scavenger hunt', 'solo music', 'group music',
'dancing competition', 'silent disco', 'mime', 'poetry
recitation', 'bonfire']
```

## Q3

```
Import random
Import bumpy as np
Import matplotlib.pyplot as plt
```

```
x=[]
y=[]
coordinates=[]
for i in range(20):
    a=random.randint(-40,40)
    b=random.randint(-40,40)
    x.append(a)
    y.append(b)
with open("clouds.txt",'w',encoding = 'utf-8',) as f:
    for i in range(0,len(x)):
        coordinates.append([x[i],y[i]])
        f.write(str(coordinates[i]))
        f.write(" \n")
plt.scatter(x,y)
plt.show()

distance = []
for i in range(len(coordinates)):
    for j in range(len(coordinates)):
        if(i !=j):
            mindist= ((coordinates[i][0] - coordinates[j][0])*2 + (coordinates[i][1]-coordinates[j][1])2)*0.5
            distance.append([coordinates[i],coordinates[j],mindist])
# for i in range(0,len(distance)):
#     # print(distance[i]," \n")
n= len(distance)
for i in range(n):
    for j in range(n - 1 - i):
        if distance[j][2] < distance[j + 1][2]:
            distance[j], distance[j + 1] = distance[j + 1], distance[j]
print(distance[n-1])
```

**answer:**

```
[[-38, -36], [33, 32], -105.0]
```

## Q2

```
Act = ["Silent Disco", "Bonfire", "Street Play", "Dancing Competition", "Short Film Screening",  
       "Rangoli", "Scavenger Hunt", "Face Art", "Solo Music", "Group Music", "Mime", "Poetry  
Recitation"]
```

```
DL = [12, 12, 2, 8, 10, 4, 4, 4, 7, 11, 2]  
Dur = [2, 2, 1, 1, 0.5, 0.5, 1.5, 0.5, 0.5, 1, 1, 1]
```

```
Sch = dict(zip(Act, list(zip(Dur, DL))))  
print(Sch)  
def scheduler(Sch):  
    sorted_Sch = {k: v for k, v in sorted(Sch.items(), key=lambda item: item[1][1])}  
    schedule = []  
    curr_time = 0  
    penal = 0  
    for i, j in sorted_Sch.items():  
        schedule.append(i)  
        print(j[1], curr_time+j[0])  
        if j[1] <= j[0]+curr_time:  
            penal+=50000*(curr_time-j[1])  
        curr_time+=j[0]  
    return schedule, abs(penal)
```

```
final, p= scheduler(Sch)  
print(final)  
print(p)
```

Penalty: 50,000