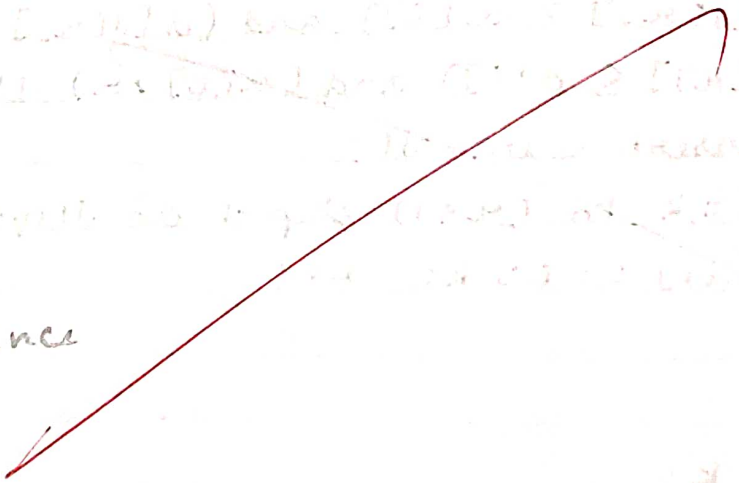
Output

```
Enter the number of jobs
7
Enter profits for each jobs
3
5
20
18
1
6
30
Enter the deadline for each job :
1
3
4
3
2
1
2
Arrange in desc :
30
20
18
6
5
3
1
The Sequence
4
1
3
2
```



The profit earned is 74

Output

```
Enter the number of nodes
6
Enter the cost matrix weights:
0   50   45   10   0   0
0   0    10   15   0   0
0   0    0    0    30  0
10  0    0    0    15  0
0   20   35   0    0   0
0   0    0    0    3   0
Enter the source vertex:
2
The shortest path from source 2 to all the
other vertices are:
source : 2    destination : 1    min cost : 25
source : 2    destination : 3    min cost : 10
source : 2    destination : 4    min cost : 25
source : 2    destination : 5    min cost 30
source : 2    destination : 6    min cost : 9999
```

Output

Enter no. of nodes in Graph 4

1 and 2 : 2
1 and 3 : 3
1 and 4 : 5
2 and 3 : 1
2 and 4 : 6
3 and 4 : 3

The min cost Edge is 2 and 3 is with cost 1
The min cost Edge is 1 and 2 is with cost 2
The min cost Edge is 4 and 1 is with cost 5
The total min cost spanning tree is 8

Output:

Enter no. of vertices in the Graph
4
Enter Edge in the Graph
4
Enter 2 vertices and weights of Edge
first vertex : 1
second vertex : 2
weight : 2
first vertex : 2
second vertex : 3
weight : 3
first vertex : 3
second vertex : 4
weight : 4
first vertex : 4
second vertex : 1
weight : 1

edge (4, 1) with weight 1 is selected
edge (1, 2) with weight 2 is selected
edge (2, 3) with weight 3 is selected
edge (3, 4) with weight 11 is discarded

Min spanning tree for
min cost : 6

Output:

```
Enter no. of vertices in the Graph
4
Enter Edge in the Graph
4
Enter 2 vertices and weights of Edge
first vertex : 1
second vertex : 2
weight : 2
first vertex : 2
second vertex : 3
weight = 3
first vertex : 3
second vertex : 4
weight = 4
first vertex : 4
second vertex : 1
weight = 1

edge (4,1) with weight 1 is selected
edge (1,2) with weight 2 is selected
edge (2,3) with weight 3 is selected
edge (3,4) with weight 4 is discarded

Min spanning tree for
min Cost = 6
```

Output

```
Enter the Number of items
7
Enter the items weights
2
3
5
7
1
4
1
Enter the items value
10
5
15
7
6
18
3
Enter the maximum capacity 15
The maximum value that can be put
in a knapsack of capacity is :54
```

Output

Optimal Binary Search tree

Enter the no. of nodes = 4

Enter the data as : : : :

a[1] = 1

a[2] = 2

a[3] = 3

a[4] = 4

p[1] = 3

p[2] = 3

p[3] = 1

p[4] = 1

q[0] = 2

q[1] = 3

q[2] = 1

q[3] = 1

q[4] = 1

The optimal Binary search tree for the Given nodes is : : : :

The Root of this OBST is 2

The cost of this OBST is 32

NODE	LEFT CHILD	RIGHT CHILD
2	1	3
1	-	-
3	-	4
4	-	-

Output

Enter the number of vertices

3

Enter the weighted Matrix of the Graph

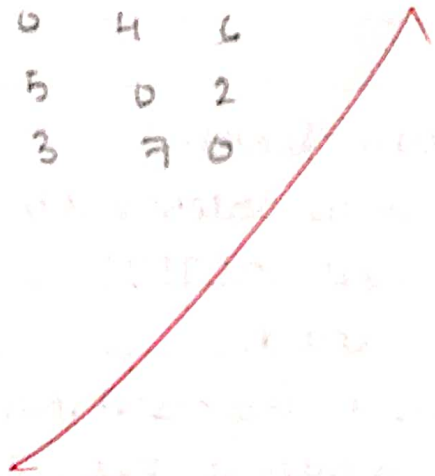
0 4 11

6 0 2

3 0 0

The transitive closure of the Graph

	1	2	3
1	0	4	6
2	5	0	2
3	3	7	0



Output

Enter no of Queens

4

Solution : 1

	1	2	3	4
1	-	Q	-	-
2	-	-	-	Q
3	Q	-	-	-
4	-	-	Q	-

Solution : 2

	1	2	3	4
1	-	-	Q	-
2	Q	-	-	-
3	-	-	-	Q