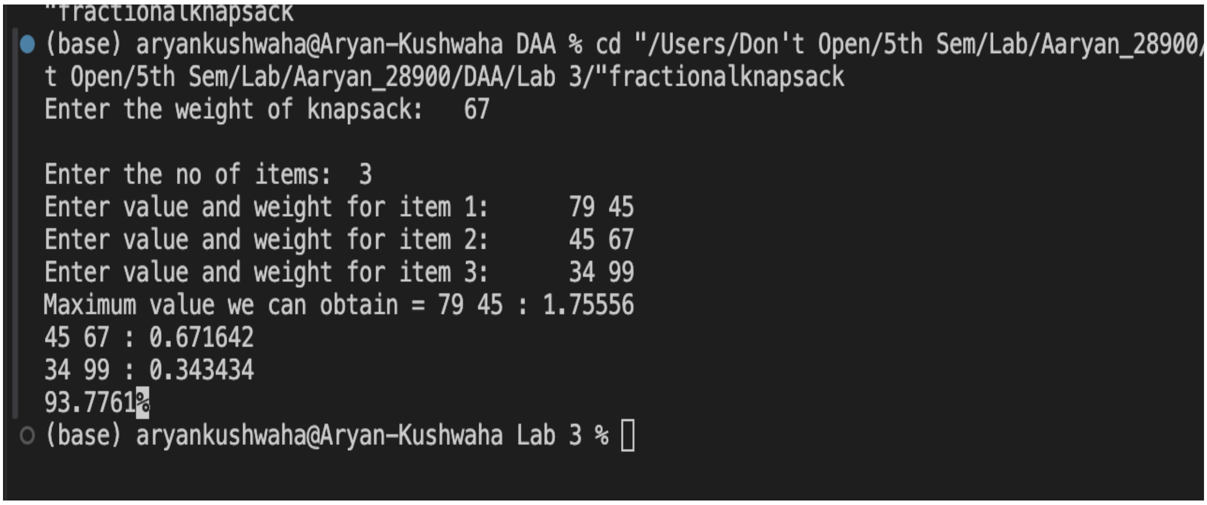
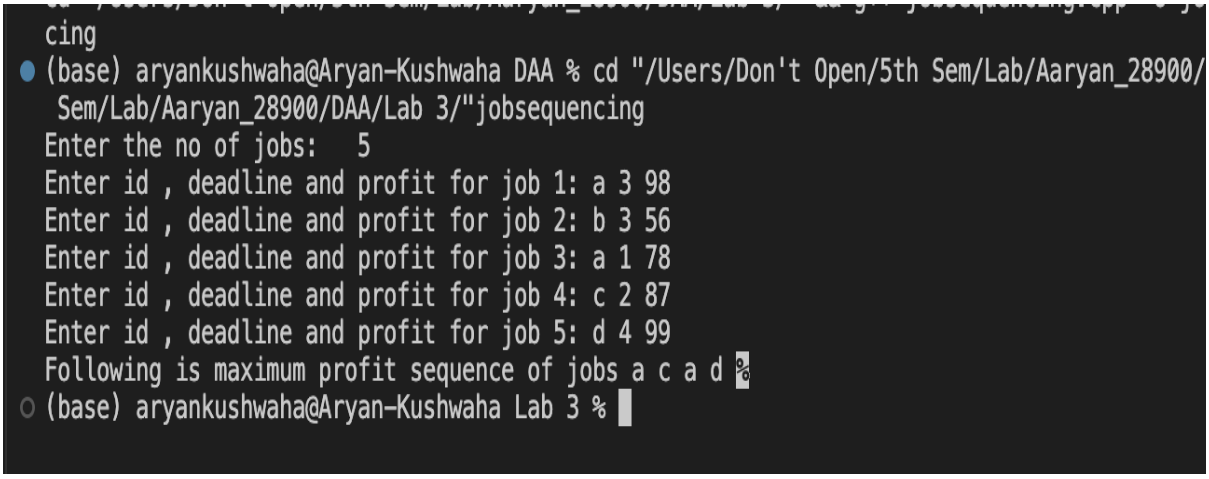
Output :- Fractional Knapsack Problem



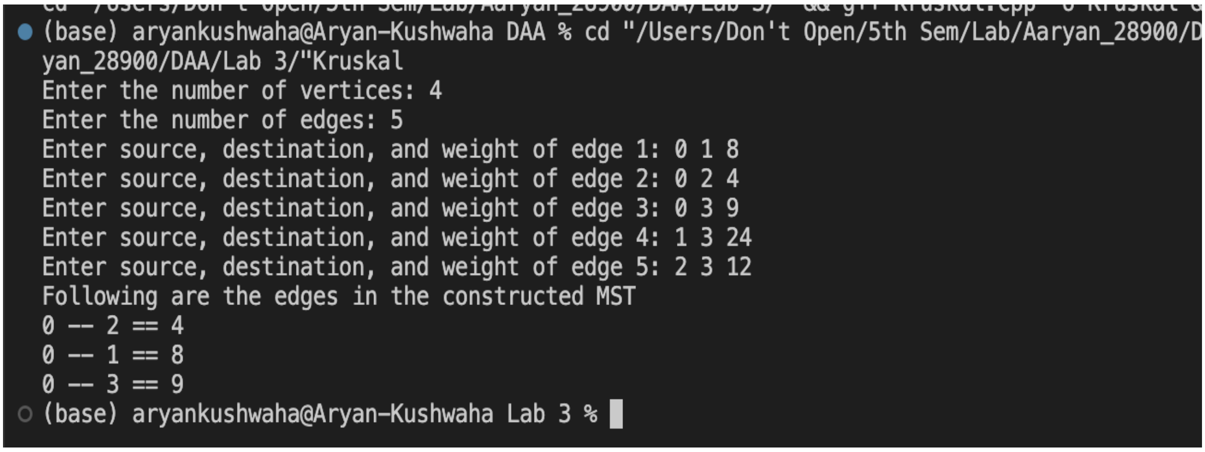
```
"fractionalknapsack
● (base) aryankushwaha@Aryan-Kushwaha DAA % cd "/Users/Don't Open/5th Sem/Lab/Aaryan_28900/
t Open/5th Sem/Lab/Aaryan_28900/DAA/Lab 3/"fractionalknapsack
Enter the weight of knapsack:   67

Enter the no of items:  3
Enter value and weight for item 1:      79 45
Enter value and weight for item 2:      45 67
Enter value and weight for item 3:      34 99
Maximum value we can obtain = 79 45 : 1.75556
45 67 : 0.671642
34 99 : 0.343434
93.7761%
○ (base) aryankushwaha@Aryan-Kushwaha Lab 3 % 
```

Output :- Job Sequencing Problem



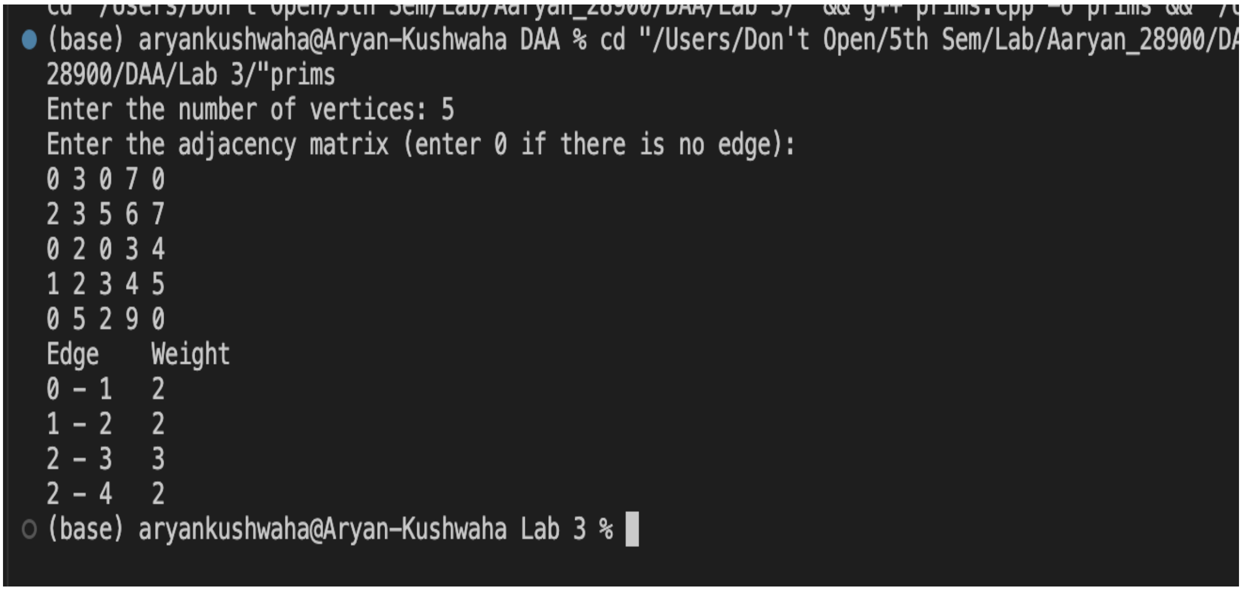
```
cing
● (base) aryankushwaha@Aryan-Kushwaha DAA % cd "/Users/Don't Open/5th Sem/Lab/Aaryan_28900/
 Sem/Lab/Aaryan_28900/DAA/Lab 3/"jobsequencing
Enter the no of jobs:   5
Enter id , deadline and profit for job 1: a 3 98
Enter id , deadline and profit for job 2: b 3 56
Enter id , deadline and profit for job 3: a 1 78
Enter id , deadline and profit for job 4: c 2 87
Enter id , deadline and profit for job 5: d 4 99
Following is maximum profit sequence of jobs a c a d %
○ (base) aryankushwaha@Aryan-Kushwaha Lab 3 % 
```

Output :- Kruskal’s Algorithm



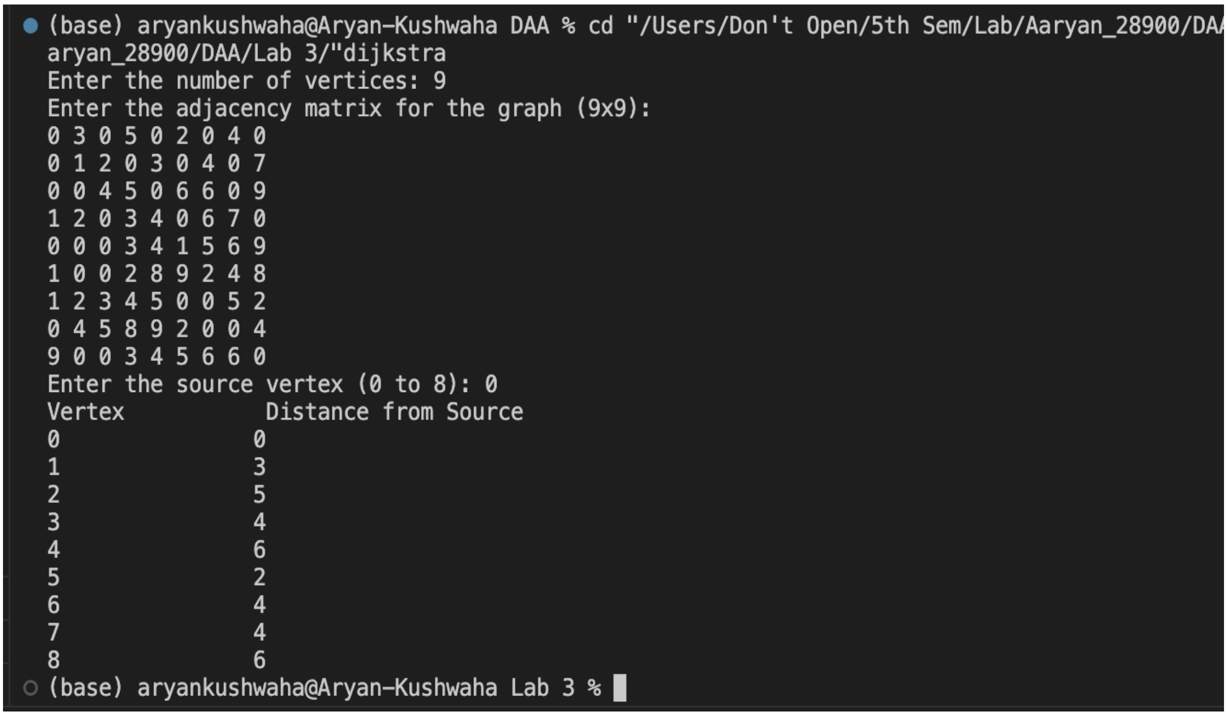
```
● (base) aryankushwaha@Aryan-Kushwaha DAA % cd "/Users/Don't Open/5th Sem/Lab/Aaryan_28900/D
yan_28900/DAA/Lab 3/"Kruskal
Enter the number of vertices: 4
Enter the number of edges: 5
Enter source, destination, and weight of edge 1: 0 1 8
Enter source, destination, and weight of edge 2: 0 2 4
Enter source, destination, and weight of edge 3: 0 3 9
Enter source, destination, and weight of edge 4: 1 3 24
Enter source, destination, and weight of edge 5: 2 3 12
Following are the edges in the constructed MST
0 -- 2 == 4
0 -- 1 == 8
0 -- 3 == 9
○ (base) aryankushwaha@Aryan-Kushwaha Lab 3 % 
```

Output :- Prim’s Algorithm



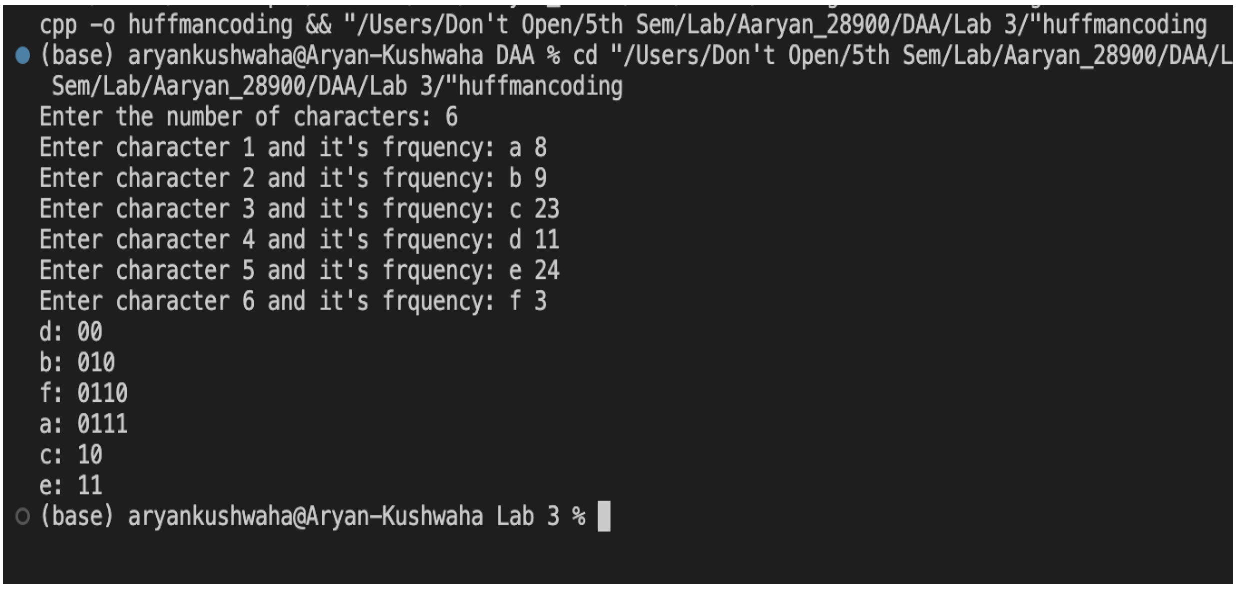
```
cd "/Users/Don't Open/5th Sem/Lab/Aaryan_28900/DAA/Lab 3/" && g++ prims.cpp -o prims && "/U
● (base) aryankushwaha@Aryan-Kushwaha DAA % cd "/Users/Don't Open/5th Sem/Lab/Aaryan_28900/DA
  28900/DAA/Lab 3/"prims
  Enter the number of vertices: 5
  Enter the adjacency matrix (enter 0 if there is no edge):
  0 3 0 7 0
  2 3 5 6 7
  0 2 0 3 4
  1 2 3 4 5
  0 5 2 9 0
  Edge    Weight
  0 - 1   2
  1 - 2   2
  2 - 3   3
  2 - 4   2
○ (base) aryankushwaha@Aryan-Kushwaha Lab 3 % █
```

Output :- Dijkstra’s Algorithm



```
● (base) aryankushwaha@Aryan-Kushwaha DAA % cd "/Users/Don't Open/5th Sem/Lab/Aaryan_28900/DA
  aryan_28900/DAA/Lab 3/"dijkstra
  Enter the number of vertices: 9
  Enter the adjacency matrix for the graph (9x9):
  0 3 0 5 0 2 0 4 0
  0 1 2 0 3 0 4 0 7
  0 0 4 5 0 6 6 0 9
  1 2 0 3 4 0 6 7 0
  0 0 0 3 4 1 5 6 9
  1 0 0 2 8 9 2 4 8
  1 2 3 4 5 0 0 5 2
  0 4 5 8 9 2 0 0 4
  9 0 0 3 4 5 6 6 0
  Enter the source vertex (0 to 8): 0
  Vertex          Distance from Source
  0               0
  1               3
  2               5
  3               4
  4               6
  5               2
  6               4
  7               4
  8               6
○ (base) aryankushwaha@Aryan-Kushwaha Lab 3 % █
```

Output :- Huffman Coding



```
 cpp -o huffmancoding && "/Users/Don't Open/5th Sem/Lab/Aaryan_28900/DAA/Lab 3/"huffmancoding
● (base) aryankushwaha@Aryan-Kushwaha DAA % cd "/Users/Don't Open/5th Sem/Lab/Aaryan_28900/DAA/L
  Sem/Lab/Aaryan_28900/DAA/Lab 3/"huffmancoding
 Enter the number of characters: 6
 Enter character 1 and it's frquency: a 8
 Enter character 2 and it's frquency: b 9
 Enter character 3 and it's frquency: c 23
 Enter character 4 and it's frquency: d 11
 Enter character 5 and it's frquency: e 24
 Enter character 6 and it's frquency: f 3
 d: 00
 b: 010
 f: 0110
 a: 0111
 c: 10
 e: 11
○ (base) aryankushwaha@Aryan-Kushwaha Lab 3 % █
```