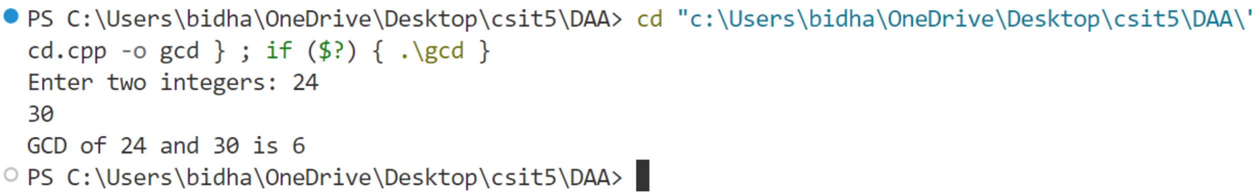
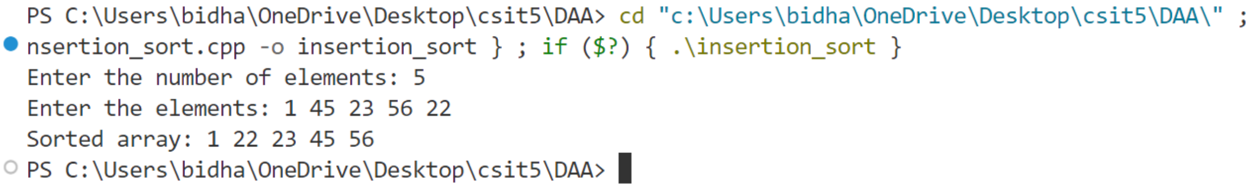
OUTPUT LAB 1



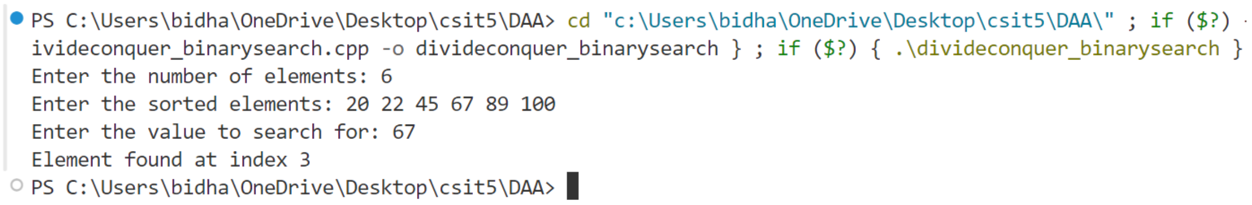
```
● PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\'
  cd.cpp -o gcd } ; if ($?) { .\gcd }
  Enter two integers: 24
  30
  GCD of 24 and 30 is 6
○ PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> █
```

OUTPUT LAB 2



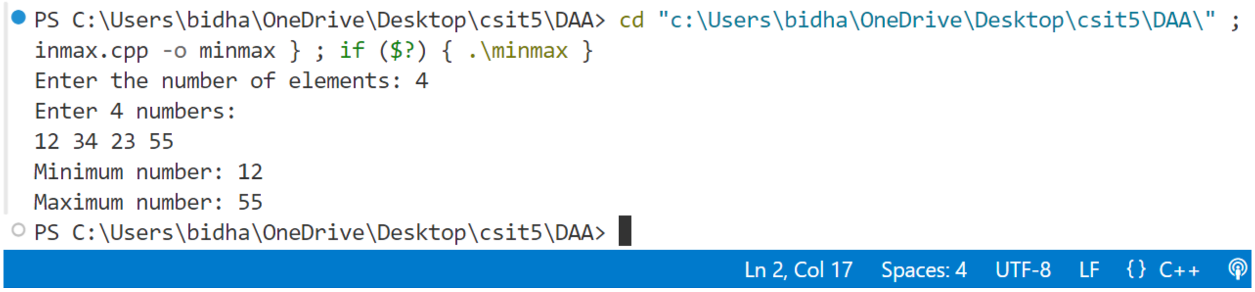
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
nsertion_sort.cpp -o insertion_sort } ; if ($?) { .\insertion_sort }
Enter the number of elements: 5
Enter the elements: 1 45 23 56 22
Sorted array: 1 22 23 45 56
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA>
```

OUTPUT LAB 3



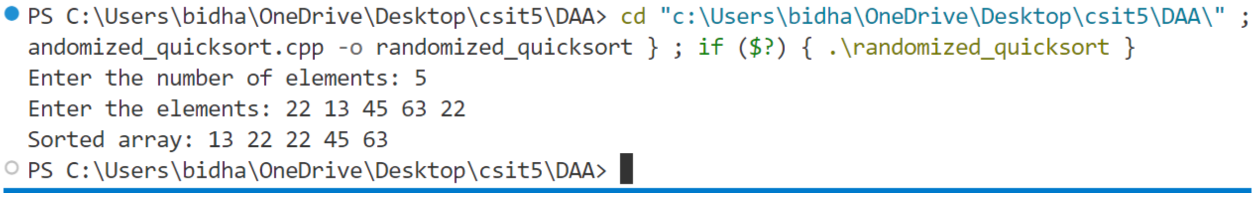
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ; if ($?) ·
ivideconquer_binarysearch.cpp -o divideconquer_binarysearch } ; if ($?) { .\divideconquer_binarysearch }
Enter the number of elements: 6
Enter the sorted elements: 20 22 45 67 89 100
Enter the value to search for: 67
Element found at index 3
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> █
```

OUTPUT LAB 4



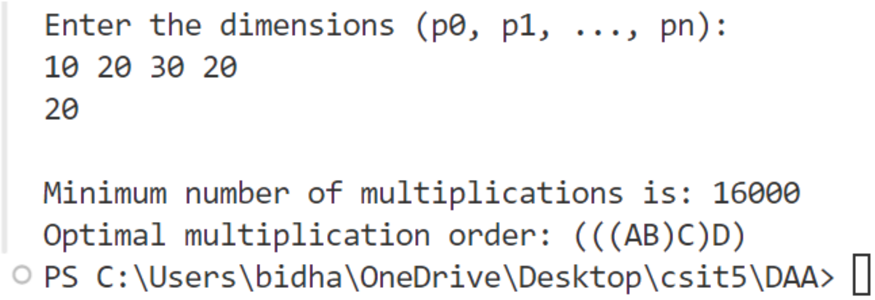
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
inmax.cpp -o minmax } ; if ($?) { .\minmax }
Enter the number of elements: 4
Enter 4 numbers:
12 34 23 55
Minimum number: 12
Maximum number: 55
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA>
```

OUTPUT LAB 5



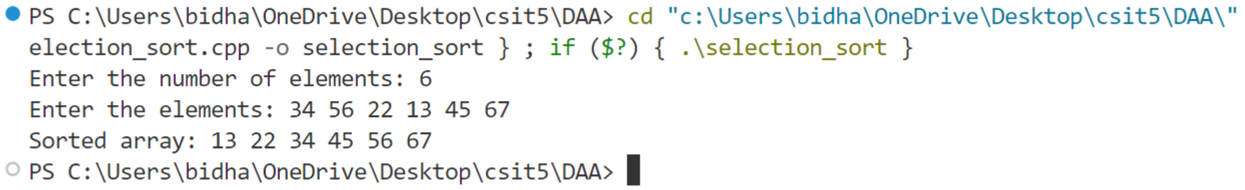
```
● PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
  andomized_quicksort.cpp -o randomized_quicksort } ; if ($?) { .\randomized_quicksort }
  Enter the number of elements: 5
  Enter the elements: 22 13 45 63 22
  Sorted array: 13 22 22 45 63
○ PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> 
```

OUTPUT LAB 6



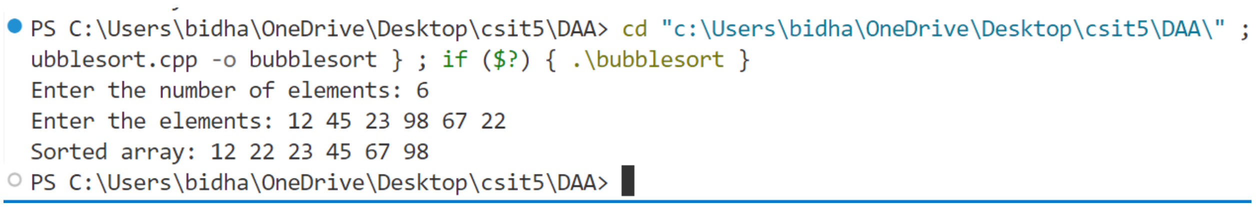
```
Enter the dimensions (p0, p1, ..., pn):
10 20 30 20
20

Minimum number of multiplications is: 16000
Optimal multiplication order: (((AB)C)D)
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA>
```

OUTPUT LAB 7



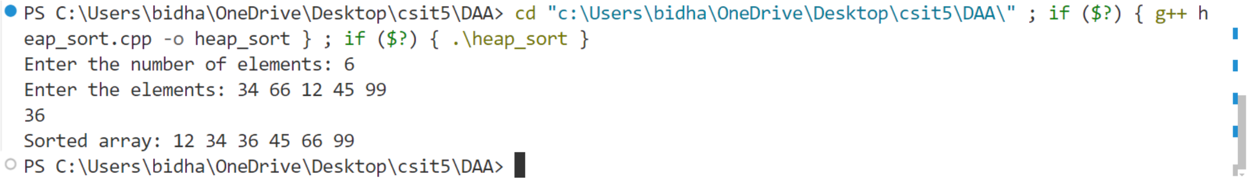
```
● PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\"
  election_sort.cpp -o selection_sort } ; if ($?) { .\selection_sort }
  Enter the number of elements: 6
  Enter the elements: 34 56 22 13 45 67
  Sorted array: 13 22 34 45 56 67
○ PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> █
```

OUTPUT LAB 8



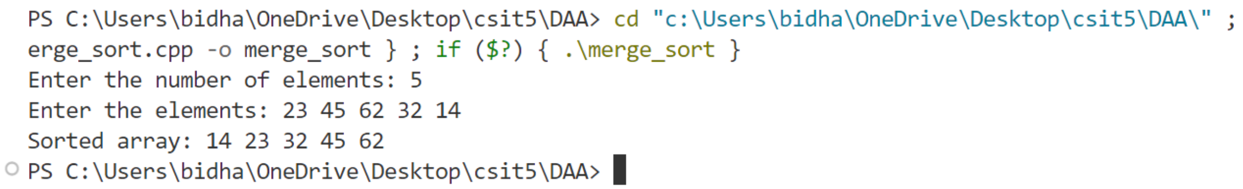
```
● PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
  ubblesort.cpp -o bubblesort } ; if ($?) { .\bubblesort }
  Enter the number of elements: 6
  Enter the elements: 12 45 23 98 67 22
  Sorted array: 12 22 23 45 67 98
○ PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> 
```

OUTPUT LAB 9



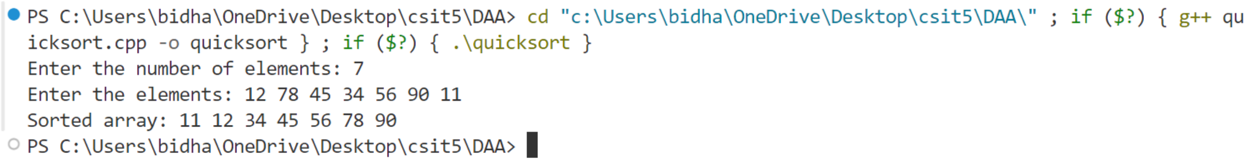
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ; if ($?) { g++ h
eap_sort.cpp -o heap_sort } ; if ($?) { .\heap_sort }
Enter the number of elements: 6
Enter the elements: 34 66 12 45 99
36
Sorted array: 12 34 36 45 66 99
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> 
```

OUTPUT LAB 10



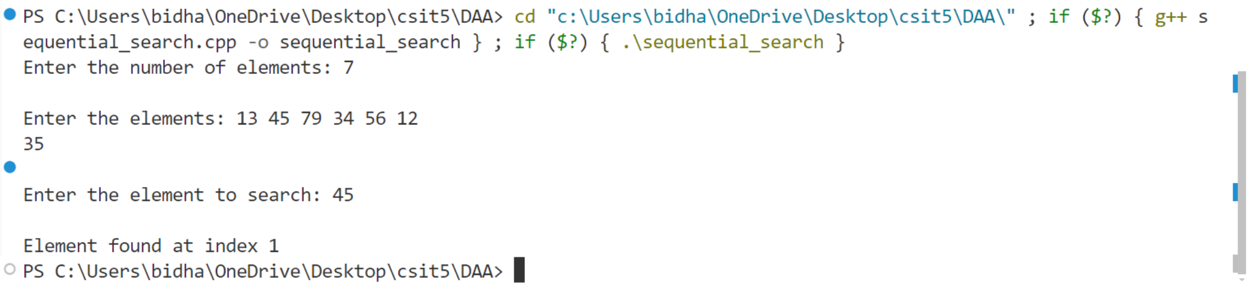
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
erge_sort.cpp -o merge_sort } ; if ($?) { .\merge_sort }
Enter the number of elements: 5
Enter the elements: 23 45 62 32 14
Sorted array: 14 23 32 45 62
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> 
```

OUTPUT LAB11



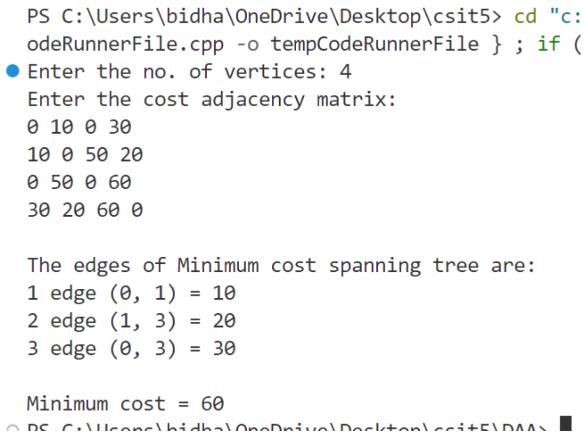
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ; if ($?) { g++ qu
icksort.cpp -o quicksort } ; if ($?) { .\quicksort }
Enter the number of elements: 7
Enter the elements: 12 78 45 34 56 90 11
Sorted array: 11 12 34 45 56 78 90
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA>
```

OUTPUT LAB 12



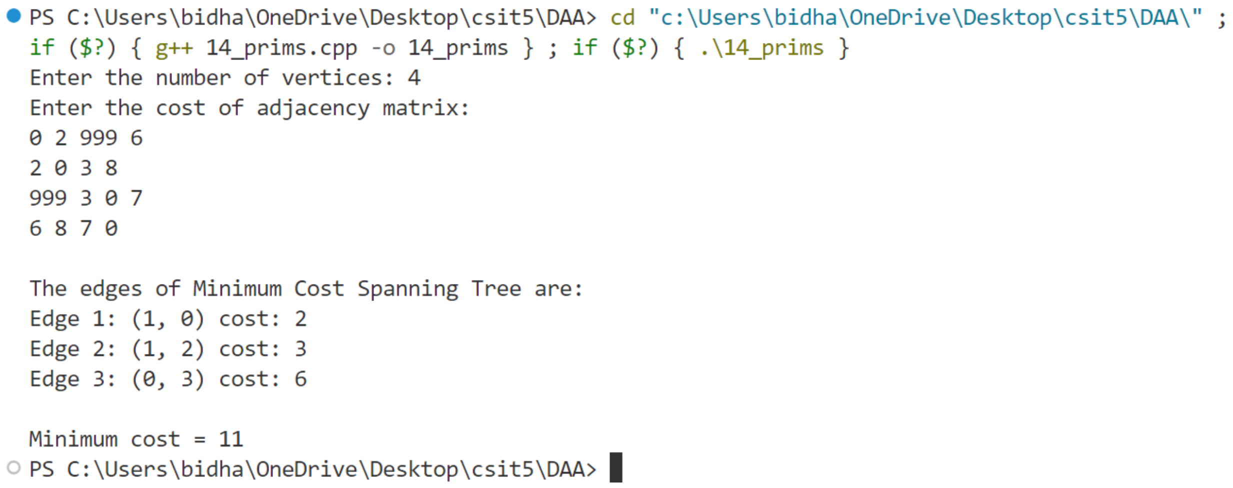
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ; if ($?) { g++ s
equential_search.cpp -o sequential_search } ; if ($?) { .\sequential_search }
Enter the number of elements: 7

Enter the elements: 13 45 79 34 56 12
35

Enter the element to search: 45

Element found at index 1
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> 
```

OUTPUT LAB 13



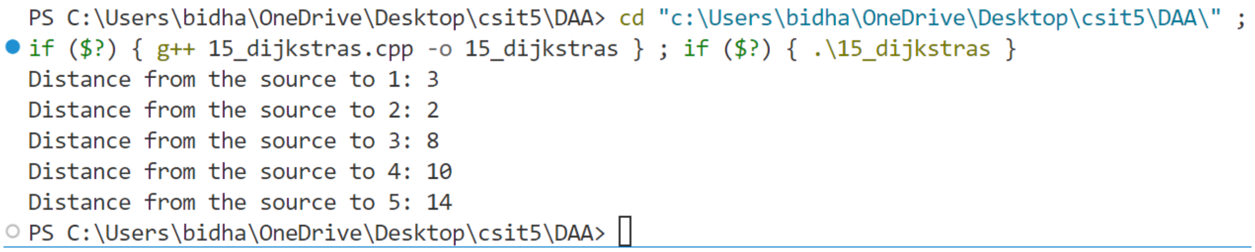
```
PS C:\Users\bidha\OneDrive\Desktop\csit5> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ; if ($?)
odeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter the no. of vertices: 4
Enter the cost adjacency matrix:
0 10 0 30
10 0 50 20
0 50 0 60
30 20 60 0

The edges of Minimum cost spanning tree are:
1 edge (0, 1) = 10
2 edge (1, 3) = 20
3 edge (0, 3) = 30

Minimum cost = 60
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA>
```

OUTPUT LAB 14



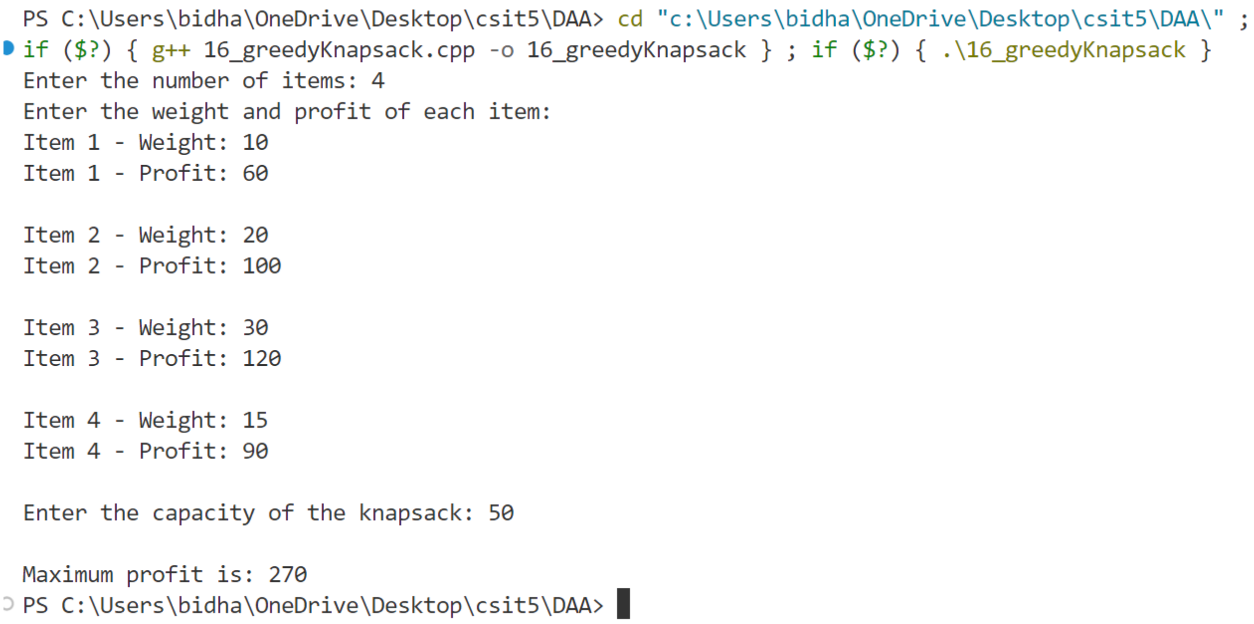
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
if ($?) { g++ 14_prims.cpp -o 14_prims } ; if ($?) { .\14_prims }
Enter the number of vertices: 4
Enter the cost of adjacency matrix:
0 2 999 6
2 0 3 8
999 3 0 7
6 8 7 0

The edges of Minimum Cost Spanning Tree are:
Edge 1: (1, 0) cost: 2
Edge 2: (1, 2) cost: 3
Edge 3: (0, 3) cost: 6

Minimum cost = 11
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA>
```

OUTPUT LAB 15



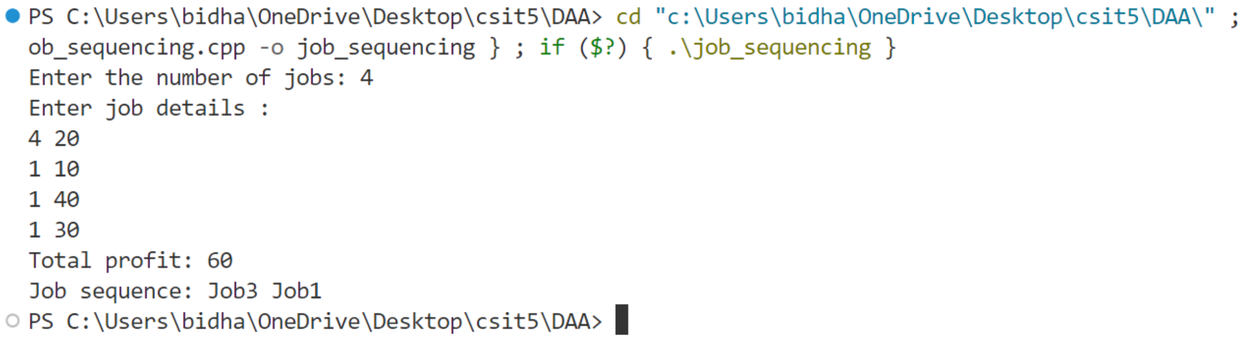
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
if ($?) { g++ 15_dijkstras.cpp -o 15_dijkstras } ; if ($?) { .\15_dijkstras }
Distance from the source to 1: 3
Distance from the source to 2: 2
Distance from the source to 3: 8
Distance from the source to 4: 10
Distance from the source to 5: 14
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> 
```

OUTPUT LAB 16



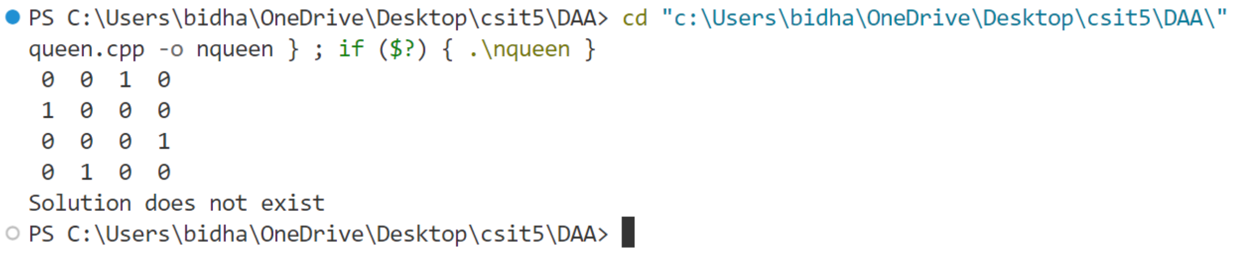
```
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
if ($?) { g++ 16_greedyKnapsack.cpp -o 16_greedyKnapsack } ; if ($?) { .\16_greedyKnapsack }
Enter the number of items: 4
Enter the weight and profit of each item:
Item 1 - Weight: 10
Item 1 - Profit: 60

Item 2 - Weight: 20
Item 2 - Profit: 100

Item 3 - Weight: 30
Item 3 - Profit: 120

Item 4 - Weight: 15
Item 4 - Profit: 90

Enter the capacity of the knapsack: 50

Maximum profit is: 270
PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA>
```

OUTPUT LAB 17



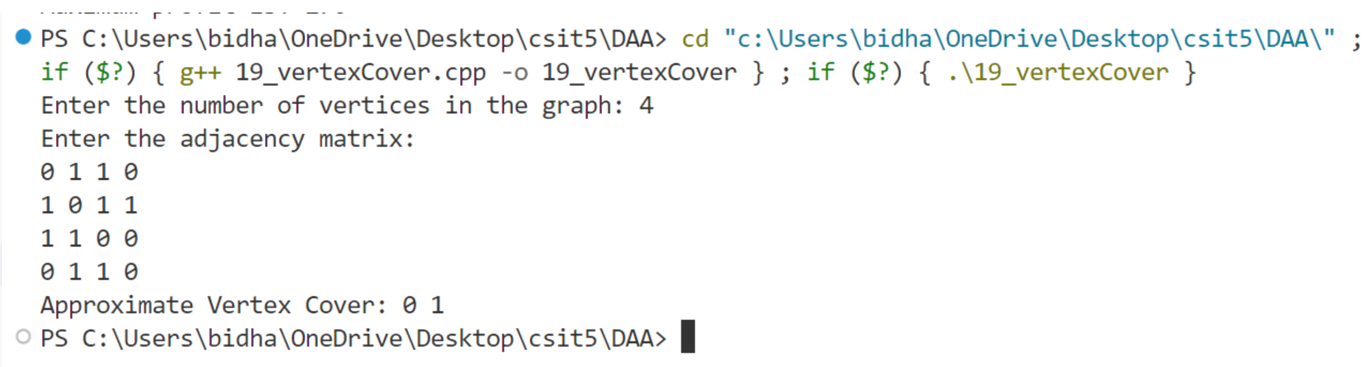
```
● PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
  ob_sequencing.cpp -o job_sequencing } ; if ($?) { .\job_sequencing }
  Enter the number of jobs: 4
  Enter job details :
  4 20
  1 10
  1 40
  1 30
  Total profit: 60
  Job sequence: Job3 Job1
○ PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> █
```

OUTPUT LAB 18



```
● PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\"
  queen.cpp -o nqueen } ; if ($?) { .\nqueen }
   0  0  1  0
   1  0  0  0
   0  0  0  1
   0  1  0  0
  Solution does not exist
○ PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> █
```

OUTPUT LAB 19



```
● PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> cd "c:\Users\bidha\OneDrive\Desktop\csit5\DAA\" ;
  if ($?) { g++ 19_vertexCover.cpp -o 19_vertexCover } ; if ($?) { .\19_vertexCover }
  Enter the number of vertices in the graph: 4
  Enter the adjacency matrix:
  0 1 1 0
  1 0 1 1
  1 1 0 0
  0 1 1 0
  Approximate Vertex Cover: 0 1
○ PS C:\Users\bidha\OneDrive\Desktop\csit5\DAA> █
```