Output

```
/tmp/zexxxVvKXz.o
Enter any number : 7
Factorial value of 7 = 5040
```

Output

```
/tmp/VLJ7MieSFy.o
How many numbers u are going to enter?: 5
Enter 5 elements: 69 50 89 12 72
Order of Sorted elements:  12 50 69 72 89
```

Output

```
/tmp/VLJ7MieSFy.o
Enter the number of elements in the Array: 5
Enter the elements:

 Array[0] = 22
 Array[1] = 13
 Array[2] = 56
 Array[3] = 100
 Array[4] = 69
 The Sorted Array is:

   13   22   56   69  100
```

Output

```
/tmp/zexxxVvKXz.o
List before sorting
10 14 19 26 27 31 33 35 42 44 0
List after sorting
0 10 14 19 26 27 31 33 35 42 44
```

Output

```
/tmp/zexxxVvKXz.o
Enter size of the list: 5
Enter any 5 integer values: 20 72 56 69 33
Enter the element to be Search: 69
Element is found at 3 index
```

```
Enter size of the list: 5
Enter any 5 integer values: 20 72 56 69 33
Enter the element to be Search: 100
Given element is not found in the list!!!
```

Output

```
/tmp/zexxxVvKXz.o
Enter size of the list: 5
Enter 5 integer values: 35 50 102 69 12
List after sorting is:  12 35 50 69 102
```

Output

```
/tmp/zexxxVvKXz.o
Enter the size of the list: 5
Enter 5 integer values in Assending order
12 25 45 69 72
Enter value to be search: 45
Element found at index 2.

Enter the size of the list: 5
Enter 5 integer values in Assending order
12 25 45 69 72

Enter value to be search: 44
Element Not found in the list.
```

Output

```
/tmp/zexxxVvKXz.o
Enter the 4 elements of first matrix: 1 2 3 4
Enter the 4 elements of second matrix: 5 6 7 8
The first matrix is

1   2
3   4
The second matrix is

5   6
7   8
After multiplication using

19  22
43  50
```

Output

```
/tmp/zexxxVvKXz.o
The minimum element is -1
The maximum element is 4
```

Output

```
/tmp/zexxxVvKXz.o
(0, 3)
(4, 4)
(3, 1)
(0, 0)
```

Output

```
/tmp/zexxxVvKXz.o
f: 0
c: 100
d: 101
a: 1100
b: 1101
e: 111
```

Output

```
/tmp/0f5AOsM0XK.o
Enter the number of items :1
Enter Weight and Profit for item[0] :
2 10
Enter the capacity of knapsack :
1
Knapsack problems using Greedy Algorithm:

The maximum value is :5.000000
```

Output

```
/tmp/pX0twFik4A.o
Preorder traversal of binary tree is
1 2 4 5 3
Inorder traversal of binary tree is
4 2 5 1 3
Postorder traversal of binary tree is
4 5 2 3 1
```

Output

```
/tmp/pX0twFik4A.o
Implementation of Kruskal's Algorithm
Enter the no. of vertices:2
Enter the cost adjacency matrix:
10
20
30
40
30

40
The edges of Minimum Cost Spanning Tree are
1 edge (1,2) =30

    Minimum cost = 30
```

Output

```
/tmp/pX0twFik4A.o
Enter no. of vertices:2
Enter the adjacency matrix:
10 20 30 40
spanning tree matrix:

0   20
20  0

Total cost of spanning tree=20S
```

Output

```
/tmp/pX0twFik4A.o
Enter 1st sequence:1256
Enter 2nd sequence:3256
The Longest Common Subsequence is 256
```

Output

```
/tmp/pX0twFik4A.o
- N Queens Problem Using Backtracking -

Enter number of Queens:1 1
Solution 1:

    1

1   Q
```

Output

```
/tmp/3P1peyZhqZ.o
Enter Number of Cities: 2
Enter Cost Matrix:

 Enter Elements of Row # : 1
1
2
Enter Elements of Row # : 2
3
4
The Cost Matrix is:


    1   2

    3   4

The Path is:

1 ===> 2 ===> 1

Minimum cost:5
```

Output

```
/tmp/3P1peyZhqZ.o
Enter number of vertices : 9
Enter edge 1( -1 -1 to quit ) : 0 1
Enter edge 2( -1 -1 to quit ) : 0 3
Enter edge 3( -1 -1 to quit ) : 0 4
Enter edge 4( -1 -1 to quit ) : 1 2
Enter edge 5( -1 -1 to quit ) : 3 6
Enter edge 6( -1 -1 to quit ) : 4 7
Enter edge 7( -1 -1 to quit ) : 6 4
Enter edge 8( -1 -1 to quit ) : 6 7
Enter edge 9( -1 -1 to quit ) : 2 5
Enter edge 10( -1 -1 to quit ) : 4 5
Enter edge 11( -1 -1 to quit ) : 7 5
Enter edge 12( -1 -1 to quit ) : 7 8
Enter edge 13( -1 -1 to quit ) : -1 -1
Enter Start Vertex for BFS:
0
0 1 3 4 2 6 5 7 8
```

Output

```
Graphs
Enter the no of edges:11
Enter the no of vertices:10
Enter the edges (format: V1 V2) : 1 2
Enter the edges (format: V1 V2) : 1 3
Enter the edges (format: V1 V2) : 2 4
Enter the edges (format: V1 V2) : 2 5
Enter the edges (format: V1 V2) : 3 6
Enter the edges (format: V1 V2) : 3 7
Enter the edges (format: V1 V2) : 4 8
Enter the edges (format: V1 V2) : 5 9
Enter the edges (format: V1 V2) : 6 10
Enter the edges (format: V1 V2) : 8 9
Enter the edges (format: V1 V2) : 9 10
0  1  1  0  0  0  0  0  0  0
 0  0  0  1  1  0  0  0  0  0
 0  0  0  0  0  1  1  0  0  0
 0  0  0  0  0  0  0  1  0  0
 0  0  0  0  0  0  0  0  1  0
 0  0  0  0  0  0  0  0  0  1
 0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  1  0
 0  0  0  0  0  0  0  0  0  1
 0  0  0  0  0  0  0  0  0  0
Enter the source: 1
1-> 2-> 4-> 8-> 9-> 10-> 5-> 3-> 6-> 7->
```

Output Clear

```
/tmp/epp4WCVdmB.o
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
    33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61
    62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
    91 92 93 94 95 96 97 98 99
```

Output

```
/tmp/wTXZ2d79wi.o
Enter some text
Thats what she said
Enter a string to find
what
Found at location: 7
```