Output

```
Enter the size of the array: 10
Enter the elements of the array: 10
9
8
7
6
5
4
3
2
1
Sorted array:
1 2 3 4 5 6 7 8 9 10


=== Code Execution Successful ===
```

Output

```
Queue elements: 10 20 30
Dequeued element: 10
Queue after one dequeue: 20 30


=== Code Execution Successful ===
```

Output

```
Enter 10 elements for the array:
1
2
3
4
5
6
7
8
9
10
Array elements are:
1   2   3   4   5   6   7   8   9   10
Menu
1. Insertion
2. Deletion
3. Linear Search
4. Binary Search
Enter your choice: 1
Enter the position to insert (0 to 10): 5
Enter the value to insert: 100
Array after insertion:
1   2   3   4   5   100 6   7   8   9   10


=== Code Execution Successful ===
```

Output

```
NTR VAL IN LIST:
4
8
9
5
6
4 -> 8 -> 9 -> 5 -> 6 -> NULL

NTR FIRST NODE IN LIST: 1
NTR VAL IN LIST:
1
4
8
9
5
6
1 -> 4 -> 8 -> 9 -> 5 -> 6 -> NULL

NTR NODE IN MIDDLE:
ntr prev node 8
ntr new node 10
NTR VAL IN LIST:
1
4
8
10
```

Output

```
10
9
5
6
1 -> 4 -> 8 -> 10 -> 9 -> 5 -> 6 -> NULL

NTR LAST NODE: 7
NTR VAL IN LIST:
1
4
8
10
9
5
6
7
1 -> 4 -> 8 -> 10 -> 9 -> 5 -> 6 -> 7 -> NULL

Ntr val to be del: 9
NTR VAL IN LIST:
1
4
8
10
5
6
```

```
Ntr val to be del: 9
NTR VAL IN LIST:
1
4
8
10
5
6
7
1 -> 4 -> 8 -> 10 -> 5 -> 6 -> 7 -> NULL


=== Code Execution Successful ===
```

Output

```
Enter the number of vertices (max 10): 4
Enter the number of edges: 4
Enter edge 1 (src dest): 2
2
Enter edge 2 (src dest): 2
3
Enter edge 3 (src dest): 4
5
Enter edge 4 (src dest): 6
7
Depth First Search starting from vertex 0:
0

=== Code Execution Successful ===
```

Output

```
Enter marks obtained in 5 subjects:
Subject 1: 40
Subject 2: 50
Subject 3: 60
Subject 4: 75
Subject 5: 65
Total marks: 290.00
Average marks: 58.00
Passed


=== Code Execution Successful ===
```