Name: Ayush garg
Rollno: 1024030878

Output upto assignment 8
ASSIGNMENT 1
Q1

```
enter no of elements in an array: 5
4
6
3
2
8
the array is: 4
6
3
2
8
enter the position at which element needs to be inserted: 2
enter the element that needs to be inserted: 76
the array after insertion is: 4
76
6
3
2
8
enter the position of element you wish to delete: 4
the array after deletion is: 4
enter the position of element you wish to delete: 4
the array after deletion is: 4
6
3
2
8
element to be found: 2
element found at position : 4
```

Q2

```
enter the no of elements: 5
enter the elements: 4

6
8
3
1
the unique and sorted array is: 1
3
4
6
8
```

Q4(A)

```
enter no of elements in an array: 5
enter the elements of array:4
7

3
9
2
the reverse array is: 2
9
3
7
4
```

Q4(C)

```
enter rows and columns of the matrix: 3
3
enter the elements of the array: 2
3
1

7
6
4

8
1
2

the transpose matrix is: 2 7 8
3 6 1
1 4 2
```

Q5

```
enter rows and columns of the matrix: 3
2
enter the elements of the array: 4
1

5
4

6
7

row 1:5
row 2:9
row 3:13
column 1:15
column 2:12
```

ASSIGNMENT 2

Q1

```
enter no of elements in an array: 5
enter the elements of array:3
9
8
11
3
element to be found: 8
element found at position : 3
```

```
enter no of elements in an array: 5
enter the elements of array:7
6
3
9
2
element to be found: 3
element found at position : 3
```

Q2

```
enter no of elements in an array: 5
enter the elements of array:65
43
27
89
48
the sorted array is 27
43
48
65
89
```

Q3

```
enter no of elements in an array: 5
enter elements of the array: 5
6
7
9
10
the missing number is at position:4
```

Q4(a-d)

```
enter the first string: hello
enter the second string: world
Concatenated string: helloworld
```

```
enter the first string: hello
the reverse string is: olleh
```

```
Enter a string: flower
String after removing vowels: flwr


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

```
Enter a character: K
Lowercase character: k


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

Q5(a-e)

```
enter the size of matrix: 3
enter diagonal elements: 4
6
7
4 0
0

0
6 0

0
0
7
```

```
Enter size of tri-diagonal matrix: 3
Enter elements of tri-diagonal matrix:
6
2
8
9
5
3
1
0
0
Matrix form: 6 2 0
9 5 3
0 0 0
```

```
Enter size of lower triangular matrix: 3
Enter elements row-wise: 3
0
0
1
8
0
3 0 0
0 0 0
1 8 0
```

```
Enter size of upper triangular matrix: 3
Enter elements row-wise: 1
5
6
0
7
8
1 5 6
0 0 7
0 0 8
```

```
Enter size of symmetric matrix: 3
Enter elements of lower triangular part row-wise: 3
5
7
9
2
1
3 5 9
5 7 2
9 2 1
```

Q8

```
enter the no of elements: 5
enter the elemetns: 77
6
32
23
52
the no of distinct elements are: 5
```

ASSIGNMENT 3

Q1

```
1. push
2. pop
3. peek
4. display
5. isfull
6. isempty
enter a choice: 1
enter a value: 4
```

Q2

```
enter the string: WEREW
WEREW

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

Q3

```
Enter an expression: city
not balanced

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

Q5

```
Enter a postfix expression (single-digit operands): 23+
Result: 5


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

ASSIGNMENT 4

Q1

```
Enter size of queue: 5
1.Enqueue
2.Dequeue
3.IsEmpty
4.IsFull
5.Display
6.Peek
Any other number to Exit
Enter your choice: 1
Enter value: 4
4 enqueued
1.Enqueue
2.Dequeue
3.IsEmpty
4.IsFull
5.Display
6.Peek
Any other number to Exit
Enter your choice: 3
Queue is not empty
```

Q2

```
Enter size of queue: 4

1.Enqueue
2.Dequeue
3.IsEmpty
4.IsFull
5.Display
6.Peek
Any other number to Exit
Enter your choice: 5
Queue is empty
```

Q3

```
Enter even number of elements: 4
Enter elements: 3
8
9
5
Interleaved queue: 3 9 8 5
```

Q5(A)

```
1.Push
2.Pop
3.Top
4.Display
5.Check Empty
Any other number to Exit
Enter your choice: 4
Stack is empty
```

Q5(b)

```
1.Push
2.Pop
3.Top
4.Display
5.Check Empty
Any other number to Exit
Enter your choice: 1
Enter value: 3
3 pushed to stack
```

ASSIGNMENT 5

Q1

```
=== Insertion Tests ===
List after insertions: 5 10 20 25 30
=== Deletion Tests ===
After deleting beginning:
10 20 25 30 After deleting end: 10 20 25 After deleting value 20: 10 25
    After deleting position 1: 10
=== Search Tests ===
Searching for 25: Not Found
Searching for 100: Not Found

=== Final List ===
10
```

Q2

```
Original list: 5 10 5 20 5
Deleted 3 occurrences of 5
Updated list: 10 20
```

Q3

```
Linked list: 10 20 30 40 50
Middle element: 30
```

Q4

```
Original list: 10 20 30 40 50
Reversed list: 50 40 30 20 10
```

AQ1

```
Intersected at 8

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

ASSIGNMENT 6

Q1

```
Element found in the Linked list!
The size of the Linked list is: 4


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

Q2

```
8 6 5 4 3 8

The size of the Circluar linked list is: 5


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

Q4

```
The Linked list contains a palindrome number!

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

Q5

```
8 6 5 4 3
The given Linked list is a Circular linked list!

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

AQ1

```
8 7 6 4 3
8 7 6
4 3


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

AQ3

```
Original list: 1 2 3 4 5 6 7 8
Reversed in groups of 3: 3 2 1 6 5 4 8 7


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

AQ4

```
Before correction: 1 2 3 (Random -> 2) 4
After correction: 1 2 3 (Random -> 4) 4


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

ASSIGNMENT 7

Q1

```
Original array: 64 34 25 12 22 11 90
Selection Sort: 11 12 22 25 34 64 90
Insertion Sort: 11 12 22 25 34 64 90
Bubble Sort: 11 12 22 25 34 64 90
Merge Sort: 11 12 22 25 34 64 90
Quick Sort: 11 12 22 25 34 64 90
```

Q2

```
Original array: 64 34 25 12 22 11 90
Sorted array: 11 12 22 25 34 64 90
```

ASSIGNMENT 8

Q1

```
1 2 3 4 5 6 7
4 2 1 3 6 5 7
1 3 2 5 7 6 4

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

Q2

```
1 2 3 4 5 6
Value found!
6
1
The inorder successor of 3 is 4


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

Q3

```
Max depth: 3
Min depth: 3


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

Q5

```
1 2 3 5 8 9
9 8 5 3 2 1

=== Code Execution Successful ==
```

Q6

```
20
10


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

AQ2

```
Total trees = 5
Tree 1: 1 null 2 null 3 null null
Tree 2: 1 null 3 2 null null null
Tree 3: 2 1 null null 3 null null
Tree 4: 3 1 null 2 null null null
Tree 5: 3 2 1 null null null null
```

AQ5

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