Data Structures and Algorithmic Techniques - CSL7561 Assignment – 1

Inputs and Outputs of Programs

Submitted by

Debonil Ghosh (M21AIE225)

1. Question no 1:
    - Input:
      AnyLongSampleString
    - Output:
      Enter a string to reverse...
      AnyLongSampleString
      Reversed string is...
      gnirtSelpmaSgnoLynA

2. Question no 2:
    - Input:
      asdfgfdsa
    - Output:
      Enter a string to check palindrome ...
      asdfgfdsa
      String is...
      Palindrome

3. Question no 3:
    - Input:
      technetronic
      ethnocentric
    - Output:
      Enter a string 1
      technetronic
      Enter a string 2
      ethnocentric
      Strings are Anagram

4. Question no 4:
    - Input:
      5
      56 45 48 25 6 35
    - Output:
      Enter the size of the array:
      5
      Enter array elements ...
      56 45 48 25 6 35
      Difference between the largest and smallest element is : 50
5. Question no 5:
    - Input & Output combined:

```
Enter the max size of the Stack:
5

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
1
Enter value to insert:
10
                    [ 10 ] pushed to stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
1
Enter value to insert:
20
                    [ 20 ] pushed to stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
1
Enter value to insert:
30
                    [ 30 ] pushed to stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
1
Enter value to insert:
40
```

```
                    [ 40 ] pushed to stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
1
Enter value to insert:
50
                    [ 50 ] pushed to stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
1
Enter value to insert:
60
                      Error : Stack Over Flowed!!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
3
Stack trace :
                      top -> 50
                             40
                             30
                             20
                             10

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
2
                    [ 50 ] poped from stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
2
                    [ 40 ] poped from stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
3
Stack trace :
```

```
                        top -> 30
                               20
                               10

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
1
Enter value to insert:
60
                     [ 60 ] pushed to stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
1
Enter value to insert:
70
                     [ 70 ] pushed to stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
3
Stack trace :
                        top -> 70
                               60
                               30
                               20
                               10

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
2
                     [ 70 ] poped from stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
2
                     [ 60 ] poped from stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
```

```
2
                    [ 30 ] poped from stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
2
                    [ 20 ] poped from stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
2
                    [ 10 ] poped from stack!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
2
                    Error : Stack empty!!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
3
Stack trace :
                      Stack empty!!

Enter
 1. to Insert (push)
 2. to Delete (pop)
 3. to See stack trace
 0. to Exit
0
```

6. Question no 6:
    - Input & Output combined:

```
Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
```

```
10
                    [ 10 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
20
                    [ 20 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
30
                    [ 30 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
40
                    [ 40 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
50
                    [ 50 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
60
                    [ 60 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
```

```
 0. to Exit
3
queue trace :
                        front ->        [ 10 ]  [ 20 ]  [ 30 ]
[ 40 ]  [ 50 ]  [ 60 ]   <-rear

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
2
                    [ 10 ] removed from Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
2
                    [ 20 ] removed from Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
2
                    [ 30 ] removed from Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
20
                    [ 20 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
3
queue trace :
                        front ->        [ 40 ]  [ 50 ]  [ 60 ]
[ 20 ]   <-rear

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
2
                    [ 40 ] removed from Queue!
```

```
Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
2
                    [ 50 ] removed from Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
2
                    [ 60 ] removed from Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
2
                    [ 20 ] removed from Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
2
                    Error : Queue empty!!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
50
                    [ 50 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
1
Enter value to insert:
100
                    [ 100 ] added to the Queue!

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
```

```
 0. to Exit
3
queue trace :
                          front ->        [ 50 ]  [ 100 ]  <-rear

Enter
 1. to Insert (enqueue)
 2. to Delete (dequeue)
 3. to See all elements
 0. to Exit
0
```

7. Question no 7:
    - Input & Output combined:

```
Enter arithmatic expression to check:
(34+23)*(78-90))
                        [ ( ] pushed to stack!
                        [ ( ] poped from stack!
                        [ ( ] pushed to stack!
                        [ ( ] poped from stack!

Arithmatic expression NOT BALANCED!

Enter arithmatic expression to check:
(((a*b)&p)+(67+90)^78)
                        [ ( ] pushed to stack!
                        [ ( ] pushed to stack!
                        [ ( ] pushed to stack!
                        [ ( ] poped from stack!
                        [ ( ] poped from stack!
                        [ ( ] pushed to stack!
                        [ ( ] poped from stack!
                        [ ( ] poped from stack!

Arithmatic expression BALANCED!
```

8. Question no 8:
    - Input & Output combined:

```
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
50
                        [ 50 ] added to the Tree!

Enter
 1. to Insert
```

```
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
60
                    [ 60 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
70
                    [ 70 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
80
                    [ 80 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
90
                    [ 90 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
```

```
1
Enter value to insert:
100
                    [ 100 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
30
                    [ 30 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
20
                    [ 20 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
40
                    [ 40 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
10
                    [ 10 ] added to the Tree!

Enter
```

```
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
15
                    [ 15 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
4
        [ 10 ]  [ 15 ]  [ 20 ]  [ 30 ]  [ 40 ]  [ 50 ]  [ 60 ]
[ 70 ]  [ 80 ]  [ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
5
        [ 50 ]  [ 30 ]  [ 20 ]  [ 10 ]  [ 15 ]  [ 40 ]  [ 60 ]
[ 70 ]  [ 80 ]  [ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
6
        [ 15 ]  [ 10 ]  [ 20 ]  [ 40 ]  [ 30 ]  [ 100 ]
[ 90 ]  [ 80 ]  [ 70 ]  [ 60 ]  [ 50 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
3
Enter value to search:
30
                    [ 30 ] found in Tree!
```

```
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
3
Enter value to search:
35
                    [ 35 ] NOT found in Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
2
Enter value to remove:
20
                    [ 20 ] found in Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
4
        [ 10 ]  [ 15 ]  [ 30 ]  [ 40 ]  [ 50 ]  [ 60 ]  [ 70 ]
[ 80 ]  [ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
2
Enter value to remove:
80
                    [ 80 ] found in Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
```

```
4
        [ 10 ]  [ 15 ]  [ 30 ]  [ 40 ]  [ 50 ]  [ 60 ]  [ 70 ]
[ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
5
        [ 50 ]  [ 30 ]  [ 10 ]  [ 15 ]  [ 40 ]  [ 60 ]  [ 70 ]
[ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
6
        [ 15 ]  [ 10 ]  [ 40 ]  [ 30 ]  [ 100 ]         [ 90 ]
[ 70 ]  [ 60 ]  [ 50 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
0
```

9. Question no 9:
    - Input & Output combined:

```
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
50
                      [ 50 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
```

```
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
60
                    [ 60 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
70
                    [ 70 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
80
                    [ 80 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
90
                    [ 90 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
```

```
Enter value to insert:
100
                    [ 100 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
30
                    [ 30 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
20
                    [ 20 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
40
                    [ 40 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
10
                    [ 10 ] added to the Tree!

Enter
 1. to Insert
```

```
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
1
Enter value to insert:
15
                    [ 15 ] added to the Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
4
        [ 10 ]  [ 15 ]  [ 20 ]  [ 30 ]  [ 40 ]  [ 50 ]  [ 60 ]
[ 70 ]  [ 80 ]  [ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
5
        [ 50 ]  [ 30 ]  [ 20 ]  [ 10 ]  [ 15 ]  [ 40 ]  [ 60 ]
[ 70 ]  [ 80 ]  [ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
6
        [ 15 ]  [ 10 ]  [ 20 ]  [ 40 ]  [ 30 ]  [ 100 ]
[ 90 ]  [ 80 ]  [ 70 ]  [ 60 ]  [ 50 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
3
Enter value to search:
30
                    [ 30 ] found in Tree!

Enter
```

```
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
3
Enter value to search:
35
                    [ 35 ] NOT found in Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
2
Enter value to remove:
20
                    [ 20 ] found in Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
4
        [ 10 ]  [ 15 ]  [ 30 ]  [ 40 ]  [ 50 ]  [ 60 ]  [ 70 ]
[ 80 ]  [ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
2
Enter value to remove:
80
                    [ 80 ] found in Tree!

Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
4
```

```
        [ 10 ]  [ 15 ]  [ 30 ]  [ 40 ]  [ 50 ]  [ 60 ]  [ 70 ]
[ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
5
        [ 50 ]  [ 30 ]  [ 10 ]  [ 15 ]  [ 40 ]  [ 60 ]  [ 70 ]
[ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
6
        [ 15 ]  [ 10 ]  [ 40 ]  [ 30 ]  [ 100 ]         [ 90 ]
[ 70 ]  [ 60 ]  [ 50 ]
Enter
 1. to Insert
 2. to Delete
 3. to Search elements
 4. to See Inorder traversal
 5. to See Preorder traversal
 6. to See Postorder traversal
 0. to Exit
0
```

10. Question no 10:

- Input & Output combined:

```
Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
1
Enter value to insert At Head:
5
                      [ 5 ] added to the List Head!
List trace :
                         -> [ 5 ] ->|

Enter
```

```
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
3
Enter value to insert At Tail:
10
                      [ 10 ] added to the List Tail!
List trace :
                         -> [ 5 ] -> [ 10 ] ->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
3
Enter value to insert At Tail:
15
                      [ 15 ] added to the List Tail!
List trace :
                         -> [ 5 ] -> [ 10 ] -> [ 15 ] ->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
3
Enter value to insert At Tail:
20
                      [ 20 ] added to the List Tail!
List trace :
                         -> [ 5 ] -> [ 10 ] -> [ 15 ] -> [ 20 ]
->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
```

```
 7. to Search
 8. to See all elements
 0. to Exit
3
Enter value to insert At Tail:
25
                      [ 25 ] added to the List Tail!
List trace :
                         -> [ 5 ] -> [ 10 ] -> [ 15 ] -> [ 20 ]
-> [ 25 ] ->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
1
Enter value to insert At Head:
1
                      [ 1 ] added to the List Head!
List trace :
                         -> [ 1 ] -> [ 5 ] -> [ 10 ] -> [ 15 ]
-> [ 20 ] -> [ 25 ] ->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
2
Enter position to insert at:
3
Enter value to insert At position [ 3 ]:
12
                      [ 12 ] added to the List at position [ 3
] !
List trace :
                         -> [ 1 ] -> [ 5 ] -> [ 10 ] -> [ 12 ]
-> [ 15 ] -> [ 20 ] -> [ 25 ] ->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
```

```
 8. to See all elements
 0. to Exit
4
                      [ 1 ] removed from the List Head!
List trace :
                         -> [ 5 ] -> [ 10 ] -> [ 12 ] -> [ 15 ]
-> [ 20 ] -> [ 25 ] ->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
5
Enter position to delete at:
2
                      [ 12 ] removed from the List position [ 2
] !
List trace :
                         -> [ 5 ] -> [ 10 ] -> [ 15 ] -> [ 20 ]
-> [ 25 ] ->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
6
                      [ 25 ] removed from the List Tail!
List trace :
                         -> [ 5 ] -> [ 10 ] -> [ 15 ] -> [ 20 ]
->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
7
Enter Value to search :
15
                    [ 15 ] found at position [ 2 ]!
List trace :
```

```
                          -> [ 5 ] -> [ 10 ] -> [ 15 ] -> [ 20 ]
->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
7
Enter Value to search :
99
                    [ 99 ] NOT found in the List!
List trace :
                          -> [ 5 ] -> [ 10 ] -> [ 15 ] -> [ 20 ]
->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
8
List trace :
                          -> [ 5 ] -> [ 10 ] -> [ 15 ] -> [ 20 ]
->|

Enter
 1. to Insert (head)
 2. to Insert (middle)
 3. to Insert (tail)
 4. to Delete (head)
 5. to Delete (middle)
 6. to Delete (tail)
 7. to Search
 8. to See all elements
 0. to Exit
0
```

11. Question no 11:
    - Input & Output combined:

Enter

```
 1. to Insert
 2. to Delete Min
 3. to See All
 0. to Exit
3
      [ 30 ]
      [ 45 ]  [ 35 ]
      [ 60 ]  [ 75 ]  [ 40 ]  [ 50 ]
      [ 70 ]  [ 65 ]  [ 80 ]  [ 95 ]  [ 90 ]
Enter
 1. to Insert
 2. to Delete Min
 3. to See All
 0. to Exit
1
Enter value to insert:
20
      [ 20 ]
      [ 45 ]  [ 30 ]
      [ 60 ]  [ 75 ]  [ 35 ]  [ 50 ]
      [ 70 ]  [ 65 ]  [ 80 ]  [ 95 ]  [ 90 ]  [ 40 ]
Enter
 1. to Insert
 2. to Delete Min
 3. to See All
 0. to Exit
2
      [ 30 ]
      [ 45 ]  [ 35 ]
      [ 60 ]  [ 75 ]  [ 40 ]  [ 50 ]
      [ 70 ]  [ 65 ]  [ 80 ]  [ 95 ]  [ 90 ]
Enter
 1. to Insert
 2. to Delete Min
 3. to See All
 0. to Exit
1
Enter value to insert:
100
      [ 30 ]
      [ 45 ]  [ 35 ]
      [ 60 ]  [ 75 ]  [ 40 ]  [ 50 ]
      [ 70 ]  [ 65 ]  [ 80 ]  [ 95 ]  [ 90 ]  [ 100 ]
Enter
 1. to Insert
 2. to Delete Min
 3. to See All
 0. to Exit
```

```
1
Enter value to insert:
120
      [ 30 ]
      [ 45 ]  [ 35 ]
      [ 60 ]  [ 75 ]  [ 40 ]  [ 50 ]
      [ 70 ]  [ 65 ]  [ 80 ]  [ 95 ]  [ 90 ]  [ 100 ]        [ 120 ]
Enter
 1. to Insert
 2. to Delete Min
 3. to See All
 0. to Exit
1
Enter value to insert:
105
      [ 30 ]
      [ 45 ]  [ 35 ]
      [ 60 ]  [ 75 ]  [ 40 ]  [ 50 ]
      [ 70 ]  [ 65 ]  [ 80 ]  [ 95 ]  [ 90 ]  [ 100 ]        [ 120 ]        [ 105 ]

Enter
 1. to Insert
 2. to Delete Min
 3. to See All
 0. to Exit
1
Enter value to insert:
25
      [ 25 ]
      [ 30 ]  [ 35 ]
      [ 45 ]  [ 75 ]  [ 40 ]  [ 50 ]
      [ 60 ]  [ 65 ]  [ 80 ]  [ 95 ]  [ 90 ]  [ 100 ]        [ 120 ]        [ 105 ]
      [ 70 ]
Enter
 1. to Insert
 2. to Delete Min
 3. to See All
 0. to Exit
1
Enter value to insert:
33
      [ 25 ]
      [ 30 ]  [ 35 ]
      [ 33 ]  [ 75 ]  [ 40 ]  [ 50 ]
      [ 45 ]  [ 65 ]  [ 80 ]  [ 95 ]  [ 90 ]  [ 100 ]        [ 120 ]        [ 105 ]
      [ 70 ]  [ 60 ]
Enter
 1. to Insert
```

2. to Delete Min
3. to See All
0. to Exit
2
[30]
[33] [35]
[45] [75] [40] [50]
[60] [65] [80] [95] [90] [100] [120] [105]
[70]
Enter
1. to Insert
2. to Delete Min
3. to See All
0. to Exit
2
[33]
[45] [35]
[60] [75] [40] [50]
[70] [65] [80] [95] [90] [100] [120] [105]

Enter
1. to Insert
2. to Delete Min
3. to See All
0. to Exit
0

12. Question no 12:
    - Input & Output combined:

Enter number of elements of the array:
10

Enter all [10] elements of the array separated by space:
50 60 80 40 55 20 25 85 35 95

Max Heap formed :
[95]
[85] [60]
[55] [80] [20] [25]
[40] [35] [50]
Enter the value of k to find the 'k'th largest element:
3

Max Heap formed after heapRemoveMin 0:
[85]
[80] [60]
[55] [50] [20] [25]

[40] [35]

Max Heap formed after heapRemoveMin 1:

[80]

[55] [60]

[40] [50] [20] [25]

[35]

The ‘k’th largest element is [80]