

## CS4097D: Object Oriented System Lab

1. Write a Java program to find whether an input element exists in the array or not ?

**Test Cases:**

**Input-** 1 2 3 4 5 6

Enter element to search 4

**Output:** Element found at 4 index.

**Input-** 1 2 3 4 5 6

Enter element to search 7

**Output:** Element not found .

2. Write a Java program to find middle element of the linked list in one pass

**Test Cases:**

**Input-** 1-> 2-> 3-> 4-> 5-> 6-> 7

**Output:** 4.

**Input-** 140-> 200-> 3-> 47-> 59-> 6

**Output:** 3

3. Write a Java program to find the third element from the end in the linkedlist in one pass

**Test Cases:**

**Input-** 1-> 2-> 3-> 4-> 5-> 6-> 7

**Output:** 5.

**Input-** 140-> 200-> 3-> 47-> 59-> 6

**Output:** 47

4. Implement stack using array in Java

**Test Cases:**

**Input-**

Push(1)

Push(2)

Pop()

Push(3)

Push(4)

Push(5)

Pop()

Push(6)

**Output:**

2

5

element in stack 1 3 4 6

**5.** Write a Java program for reversing a queue by using only standard operations enqueue(x) and dequeue()

**Test Cases:**

**Input :** Q = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

**Output :** Q = [100, 90, 80, 70, 60, 50, 40, 30, 20, 10]

**Input :** [1, 2, 3, 4, 5]

**Output :** [5, 4, 3, 2, 1]

**6.** Given an expression string **exp**. Examine whether the pairs and the orders of “{”, “}”, “(”, “)”, “[”, “]” are correct in exp.

For example, the program should print 'balanced' for exp = “[()]{}{[()()]()}” and 'not balanced' for exp = “[()]”

**Test Cases:**

**Input:**{([])} **Output:** balanced

**Input:**() **Output:** balanced

**Input:**{([]) **Output:** not balanced

**7.** You are given N elements and your task is to Implement a Stack in which you can get minimum element in O(1) time.

**Input:**

The first line of the input contains an integer 'T' denoting the number of test cases. Then T test cases follow. First line of each test case contains an integer Q denoting the number of queries.

A Query Q may be of 3 Types:

1. 1 x (a query of this type means pushing 'x' into the stack)
2. 2 (a query of this type means to pop element from stack and print the popped element)
3. 3 (a query of this type means to print the minimum element from the stack)

The second line of each test case contains Q queries separated by space.

**Output:**

The output for each test case will be space separated integers having -1 if the stack is empty else the element popped out or min element from the stack.

**User Task:**

You are required to complete the three methods push() which take one argument an integer 'x' to be pushed into the stack, pop() which returns a integer popped out from the stack and getMin() which returns the min element from the stack.

**Test Cases:****Input:**

1

6

1 2 1 3 2 3 1 1 3

**Output:**

3 2 1

**Explanation:**

Testcase 1:

In the first test case for query

1 2 the stack will be {2}

1 3 the stack will be {2 3}

2 popped element will be 3 the stack will be {2}

3 min element will be 2

1 1 the stack will be {2 1}

3 min element will be 1

8. Given an array arr of N integers. Find the contiguous sub-array with maximum sum.

**Input:**

The first line of input contains an integer T denoting the number of test cases. The description of T test cases follows. The first line of each test case contains a single integer N denoting the size of array. The second line contains N space-separated integers A1, A2, ..., AN denoting the elements of the array.

**Output:**

Print the maximum sum of the contiguous sub-array in a separate line for each test case.

**Example:****Input**

2

5  
1 2 3 -2 5  
4  
-1 -2 -3 -4

**Output**

9  
-1

**Explanation:**

Testcase 1: Max subarray sum is 9 of elements (1, 2, 3, -2, 5) which is a contiguous subarray.

9. Given a matrix `mat[][]` of size  $M \times N$ . Traverse and print the matrix in spiral form.

**Input:**

The first line of the input contains a single integer  $T$ , denoting the number of test cases. Then  $T$  test cases follow. Each test case has 2 lines. First line contains  $M$  and  $N$  respectively separated by a space. Second line contains  $M \times N$  values separated by spaces.

**Output:**

Elements when travelled in Spiral form, will be displayed in a single line.

**Test Cases:**

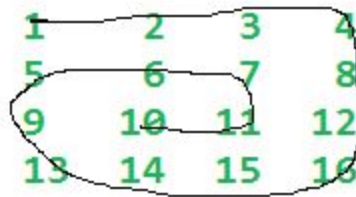
**Input:**

2  
4 4  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16  
3 4  
1 2 3 4 5 6 7 8 9 10 11 12

**Output:**

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10  
1 2 3 4 8 12 11 10 9 5 6 7

**Input :**



**Output :**

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

10. Given an array of N strings, find the longest common prefix among all strings present in the array.

**Input:**

The first line of the input contains an integer T which denotes the number of test cases to follow. Each test case contains an integer N. Next line has space separated N strings.

**Output:**

Print the longest common prefix as a string in the given array. If no such prefix exists print "-1"(without quotes).

**Test Cases:**

**Input:**

2

4

geeksforgeeks geeks geek geezer

3

apple ape april

**Output:**

gee

ap