

**Instructions:**

**Total:- 10 Marks**

1. Solve any 9 questions.
2. Input should be from user.
3. Indentation and comments mandatory.
4. Each program 1 Marks and all comments 1 Marks.
5. Do not use any inbuilt functions.

**Q1. Write a java program to print 1 to nth Strong number.**

**Q2. Write a java program to print this pattern.**

```
          *
        *   *   *
      *   *       *
    *           *
  *   *   *   *   *   *   *
*           *
  *       *
    *   *
      *   *
    *   *
```

**Q3. Write a java program to Check If a Number Is a Neon Number or Not Neon number using function recursion.**

**Q4. Problem:**

Given an array of positive integers and a target sum S, find the length of the smallest contiguous subarray whose sum is greater than or equal to S. If no such subarray exists, return 0.

**Example:**

**Input: arr = [2,3,1,2,4,3], S = 7**

**Output: 2 (subarray [4,3]).**

**Explanation:**

Expand window until the sum  $\geq S$ . Then shrink from the left while maintaining the condition to minimize length.

**Q5. Write a java program to find the “ Leaders ” in an array.**

- A leader in an array is an element which is larger than all the elements to its right side, which is what you can see in the above output.

**Input :- arr – { 15 , 18 , 5 , 7 , 9 , 2 }**

**Output :- 2 , 9 , 18**

**Q6.** Create class name as ArrayOperation with method name as setArray() and create its Two child classes name as CeilFloor , Matrix. We need to inherit the ArrayOperation class in CeilFloor, Matrix and create method. and write the logic.

**1. CeilFloor Class :-**

**Expected Output :**

```
The given array is : 1 3 5 7 8 9  
Number: 0 ceiling is: 1 floor is: -1  
Number: 1 ceiling is: 1 floor is: 1  
Number: 2 ceiling is: 3 floor is: 1  
Number: 3 ceiling is: 3 floor is: 3  
Number: 4 ceiling is: 5 floor is: 3  
Number: 5 ceiling is: 5 floor is: 5  
Number: 6 ceiling is: 7 floor is: 5  
Number: 7 ceiling is: 7 floor is: 7  
Number: 8 ceiling is: 8 floor is: 8  
Number: 9 ceiling is: 9 floor is: 9  
Number: 10 ceiling is: -1 floor is: 9
```

**2. Matrix class :-**

**Enter 9 elements for the 3x3 matrix:**

```
10 25 40  
50 15 20  
30 35 45
```

**Expected Output :**

```
Second max in column 1: 30  
Second max in column 2: 25  
Second max in column 3: 40
```

**Q7. Problem Statement:** Create an abstract class Student with attributes roll number, name, and an array of marks (5 subjects).

Create an interface ResultOperations with methods calculateTotal(), calculatePercentage(), and assignGrade().

- Implement UGStudent and PGStudent classes with grading rules:

```
UG: Pass if percentage ≥ 40%  
PG: Pass if percentage ≥ 50%
```

**Additional Requirements:**

1. Store details for N students in an array.

2. Display:

- List of passed and failed students separately.
- Top 3 students by percentage.
- Average marks in each subject.

**Explanation:** Covers abstraction for common structure, interface for calculations, array processing for N students, sorting for top students, and subject-wise aggregation.

**Q8. Write a program that maintains a Vector of city names. Perform the following:**

1. Insert 5 cities.
2. Remove the city at index 3.
3. Insert a new city at index 1.
4. Display final list.

**Q9. Write a program using ArrayList to store exam marks of students. Take a number from the user and count how many times it appears in the list.**

**Explanation:**

- Store marks in an ArrayList.
- Traverse the list using a loop.
- Compare each element with the user input and maintain a count.
- Demonstrates searching and frequency counting using ArrayList.

**Q10. Problem: Store integers in a Vector. Remove all odd numbers and print only even numbers.**

-----ALL THE BEST-----