

Rajalakshmi Engineering College

Name: Elamvazhuthi MS

Email: 240701133@rajalakshmi.edu.in

Roll no: 240701133

Phone: 7010282287

Branch: REC

Department: CSE - Section 4

Batch: 2028

Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 4_CY

Attempt : 1

Total Mark : 40

Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

In a university library, librarians need to track the usage of special characters in students' notes.

To help them, you are asked to write a program that counts the number of specific symbols in each passage of text.

The symbols of interest are:

Exclamation marks (!) Colons (:) Semicolons (;

Input Format

The first line of input contains an integer T, representing the number of test cases (passages).

Each of the next T lines contains a single passage of text.

Output Format

For each test case, print three integers separated by spaces, representing the number of exclamation marks, colons, and semicolons in the passage.

The first line of output corresponds to the first passage, the second line to the second passage, and so on.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
Hello! How are you
Output: 1 0 0

Answer

```
// You are using Java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = Integer.parseInt(sc.nextLine());

        for (int t = 0; t < T; t++) {
            String passage = sc.nextLine();
            int exclamation = 0, colon = 0, semicolon = 0;

            for (int i = 0; i < passage.length(); i++) {
                char ch = passage.charAt(i);
                if (ch == '!') exclamation++;
                else if (ch == ':') colon++;
                else if (ch == ';') semicolon++;
            }

            System.out.println(exclamation + " " + colon + " " + semicolon);
        }
    }
}
```

```
}
```

Status : Correct

Marks : 10/10

2. Problem Statement

Riya is preparing for a vocabulary test. Her teacher told her to focus on long words in her practice sentences, specifically words that have at least 5 letters.

Riya wants to write a program that will help her identify such words quickly.

Your task is to help Riya by printing all the words in a given sentence that have a length greater than or equal to 5.

If no such word exists, display "No long words found".

Input Format

The input contains a single line containing a sentence with multiple words.

Output Format

The output prints all words having length ≥ 5 , separated by a space.

If no such word is found, print "No long words found".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: The quick brown fox jumps over the lazy dog

Output: quick brown jumps

Answer

```
// You are using Java
import java.util.*;

class Main {
    public static void main(String[] args) {
```

```
Scanner sc = new Scanner(System.in);
String sentence = sc.nextLine();

String[] words = sentence.split(" ");
ArrayList<String> longWords = new ArrayList<>();

for (String word : words) {
    if (word.length() >= 5) {
        longWords.add(word);
    }
}

if (longWords.isEmpty()) {
    System.out.println("No long words found");
} else {
    System.out.println(String.join(" ", longWords));
}
```

Status : Correct

Marks : 10/10

3. Problem Statement

In a college, students are required to create unique usernames for accessing the digital library.

The librarian needs your help to verify whether the usernames entered by students are valid.

A username is considered valid if:

It contains only letters (a–z, A–Z) and digits (0–9). Its length is between 5 and 15 characters (inclusive). It must start with a letter (not a digit).

Your task is to determine whether each username in the list is valid or not.

Input Format

The first line of input contains an integer T, representing the number of usernames to check.

The next T lines each contain a string S, representing a username.

Output Format

For each username S, the output print "YES" if it is valid.

Otherwise, the output print "NO".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1

Alice123

Output: YES

Answer

```
// You are using Java
import java.util.*;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = Integer.parseInt(sc.nextLine());

        for (int t = 0; t < T; t++) {
            String username = sc.nextLine();
            if (isValid(username)) {
                System.out.println("YES");
            } else {
                System.out.println("NO");
            }
        }
    }

    public static boolean isValid(String username) {
        if (username.length() < 5 || username.length() > 15) return false;
        char first = username.charAt(0);
        if (!Character.isLetter(first)) return false;
```

```
for (int i = 0; i < username.length(); i++) {  
    char ch = username.charAt(i);  
    if (!Character.isLetterOrDigit(ch)) return false;  
}  
  
return true;  
}  
}
```

Status : Correct

Marks : 10/10

4. Problem Statement

A bookstore wants to analyze the titles of books to determine their longest word in each title. This helps in designing banners and covers.

Your task is to write a program that, given a sentence (book title), finds and prints the longest word. If multiple words have the same maximum length, print the first one.

Input Format

The input contains a single line containing a sentence representing the book title.

Output Format

The output prints a string representing the longest word in the sentence (book title).

Refer to the sample output for formatting specifications.

Sample Test Case

Input: The Chronicles of Narnia

Output: Chronicles

Answer

```
// You are using Java  
import java.util.*;
```

```
class Main {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        String sentence = sc.nextLine();  
  
        String[] words = sentence.split(" ");  
        String longest = "";  
        int maxLen = 0;  
  
        for (String word : words) {  
            if (word.length() > maxLen) {  
                maxLen = word.length();  
                longest = word;  
            }  
        }  
        System.out.println(longest);  
    }  
}
```

Status : Correct

Marks : 10/10