Rajalakshmi Engineering College

Name: Prajeet V
Email: 240701389@rajalakshmi.edu.in
Roll no: 240701389
Phone: 9363389322
Branch: REC
Department: CSE - Section 8
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

Anjali is preparing a report on text complexity. She wants to identify all words in a sentence that contain at least one digit so she can analyze numeric mentions.

Your task is to write a program that extracts and prints all words containing at least one digit from a given sentence.

If no such word exists, print "No words with digits found".

Input Format

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

Output Format

The output prints all words containing at least one digit separated by a space.

If no word contains a digit, print "No words with digits found".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: The model X100 and Y200 are available
Output: X100 Y200

Answer

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String line = sc.nextLine();
        String[] words = line.split(" ");
        StringBuilder result = new StringBuilder();
        for (String word : words) {
            if (word.matches(".*\\d.*")) {
                result.append(word).append(" ");
            }
        }
        if (result.length() > 0) {
            System.out.println(result.toString().trim());
        } else {
            System.out.println("No words with digits found");
        }
    }
}
```

Status : Correct ***Marks : 10/10***

2. 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 contais 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.Scanner;

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

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.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = Integer.parseInt(sc.nextLine());
        for (int i = 0; i < T; i++) {
            String username = sc.nextLine();
            if (username.length() >= 5 && username.length() <= 15 &&
                Character.isLetter(username.charAt(0)) &&
                username.matches("[a-zA-Z0-9]+")) {
                System.out.println("YES");
            } else {
                System.out.println("NO");
            }
        }
    }
}
```

Status : Correct **Marks : 10/10**

4. 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.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = Integer.parseInt(sc.nextLine());
        for (int i = 0; i < T; i++) {
            String passage = sc.nextLine();
            int exclamation = 0, colon = 0, semicolon = 0;
            for (int j = 0; j < passage.length(); j++) {
                char ch = passage.charAt(j);
                if (ch == '!') exclamation++;
                else if (ch == ':') colon++;
                else if (ch == ';') semicolon++;
            }
            System.out.println(exclamation + " " + colon + " " + semicolon);
        }
    }
}
```

Status : Correct ***Marks : 10/10***