Rajalakshmi Engineering College

Name: Prince Rohith
Email: 240701399@rajalakshmi.edu.in
Roll no: 240701399
Phone: 6369941431
Branch: REC
Department: CSE - Section 5
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

Neha is analyzing text messages to identify words that have repeated characters. A word is considered “repetitive” if any character appears more than once in that word.

Your task is to write a program that extracts all words that contain repeated characters from a given sentence.

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

Input Format

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

Output Format

The output prints all words that contain repeated characters separated by a space.

If no word contains repeated characters, print "No repetitive words found".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: letter balloon apple tree

Output: letter balloon apple tree

Answer

```
import java.util.Scanner;
import java.util.HashSet;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String sentence = sc.nextLine();
        String[] words = sentence.split(" ");
        StringBuilder result = new StringBuilder();
        boolean found = false;
        for (String word : words) {
            if (hasRepeatedChar(word)) {
                result.append(word).append(" ");
                found = true;
            }
        }
        if (found) {
            System.out.println(result.toString().trim());
        } else {
            System.out.println("No repetitive words found");
        }
    }

    public static boolean hasRepeatedChar(String word) {
        HashSet<Character> seen = new HashSet<>();
        for (char ch : word.toCharArray()) {
            if (seen.contains(ch)) {
```

```
                return true;
            }
            seen.add(ch);
        }
        return false;
    }
}
```

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

2. Problem Statement

A library wants to analyze book titles to count the number of words that start with an uppercase letter. This helps the library track proper nouns and important words in titles.

Your task is to write a program that, for each given title, counts and prints the number of words that start with an uppercase letter.

Input Format

The first line contains an integer T, representing the number of book titles.

Each of the next T lines contains a single title (string).

Output Format

For each title, the output print a single integer representing the number of words starting with an uppercase letter.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
The Chronicles of Narnia
Output: 3

Answer

```
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 title = sc.nextLine();
            int count = countWordsStartingWithUppercase(title);
            System.out.println(count);
        }
        sc.close();
    }

    public static int countWordsStartingWithUppercase(String title) {
        String[] words = title.split(" ");
        int count = 0;
        for (String word : words) {
            if (word.length() > 0 && Character.isUpperCase(word.charAt(0))) {
                count++;
            }
        }
        return count;
    }
}
```

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

3. 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

```
import java.util.Scanner;

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

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

```
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 exclamationCount = 0;
         int colonCount = 0;
```

```
        int semicolonCount = 0;
        for (char ch : passage.toCharArray()) {
            if (ch == '!') {
                exclamationCount++;
            } else if (ch == ':') {
                colonCount++;
            } else if (ch == ';') {
                semicolonCount++;
            }
        }
        System.out.println(exclamationCount + " " + colonCount + " " +
semicolonCount);
    }
    sc.close();
  }
}
```

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