Rajalakshmi Engineering College

Name: Jai Shuriya J
Email: 240701200@rajalakshmi.edu.in
Roll no: 240701200
Phone: 8754381941
Branch: REC
Department: CSE - Section 10
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

Meera is practicing her English vocabulary. She wants to focus on words that have more vowels in them, as they help improve her pronunciation. She decides to extract only those words from a sentence that contain at least two vowels.

Your task is to help Meera by writing a program that finds such words from the given sentence.

Input Format

The input contains a string representing the sentence.

Output Format

The output prints all the words that contain at least two vowels, separated by a space.

If no such word exists, print "No words with two vowels".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: This is an example sentence
Output: example sentence

Answer

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String sentence = sc.nextLine().trim();
        sc.close();

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

        for (String word : words) {
            if (countVowels(word) >= 2) {
                result.add(word);
            }
        }

        if (result.isEmpty()) {
            System.out.println("No words with two vowels");
        } else {
            System.out.println(String.join(" ", result));
        }
    }

    private static int countVowels(String word) {
        int count = 0;
        for (char ch : word.toCharArray()) {
            char lower = Character.toLowerCase(ch);
            if (lower == 'a' || lower == 'e' || lower == 'i' || lower == 'o' || lower == 'u') {
                count++;
            }
        }
        return count;
    }
}
```

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

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

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

        for (int t = 0; t < T; t++) {
            String title = sc.nextLine().trim();
            String[] words = title.split(" ");
            int count = 0;

            for (String word : words) {
                if (!word.isEmpty() && Character.isUpperCase(word.charAt(0))) {
                    count++;
                }
            }
            System.out.println(count);
        }
        sc.close();
    }
}
```

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

3. 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.*;

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

      for (int t = 0; t < T; t++) {
         String passage = sc.nextLine();
         int exclamations = 0, colons = 0, semicolons = 0;

         for (char ch : passage.toCharArray()) {
            if (ch == '!') {
               exclamations++;
            } else if (ch == ':') {
               colons++;
            } else if (ch == ';') {
               semicolons++;
            }
         }
      System.out.println(exclamations + " " + colons + " " + semicolons);
    }

    sc.close();
  }
}
```

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

4. 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

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String sentence = sc.nextLine().trim();
        sc.close();

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

        for (String word : words) {
            if (hasRepeatedChar(word)) {
                repetitiveWords.add(word);
            }
        }

        if (repetitiveWords.isEmpty()) {
            System.out.println("No repetitive words found");
        } else {
            System.out.println(String.join(" ", repetitiveWords));
        }
    }

    private static boolean hasRepeatedChar(String word) {
        Set<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***