Rajalakshmi Engineering College

Name: RAKSHITHA R
Email: 240701418@rajalakshmi.edu.in
Roll no: 240701418
Phone: 7305274265
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_PAH

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Ravi is analyzing text messages for his research on typing patterns. He wants to count the number of uppercase letters, lowercase letters, and digits in a sentence to understand typing trends.

Your task is to help Ravi by writing a program that takes a sentence and prints the count of uppercase letters, lowercase letters, and digits.

Input Format

The input contains a single line containing a sentence (string).

Output Format

The output prints three integers separated by spaces:

- Number of uppercase letters
- Number of lowercase letters
- Number of digits

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Hello World 123

Output: 2 8 3

Answer

```
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String sentence = sc.nextLine();
        int upper = 0, lower = 0, digits = 0;

        for (char ch : sentence.toCharArray()) {
            if (Character.isUpperCase(ch)) {
                upper++;
            } else if (Character.isLowerCase(ch)) {
                lower++;
            } else if (Character.isDigit(ch)) {
                digits++;
            }
        }
        System.out.println(upper + " " + lower + " " + digits);
    }
}
```

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

2. Problem Statement

At a digital library, the system needs to analyze passages to identify the frequency of vowels, since they are key for linguistic research. You are

asked to write a program that counts the number of vowels in each passage of text.

The vowels of interest are:

a, e, i, o, u (both uppercase and lowercase).

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 a single integer representing the total number of vowels 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 World
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 passage = sc.nextLine();
         int count = 0;
```

```
            for (char ch : passage.toCharArray()) {
                ch = Character.toLowerCase(ch);
                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                    count++;
                }
            }
            System.out.println(count);
        }
        sc.close();
    }
}
```

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

3. Problem Statement

Riya is preparing a puzzle game for her friends. She wants to include a feature that highlights special words in a sentence — specifically, palindromic words (words that read the same forward and backward).

Your task is to help Riya by writing a program that extracts all palindrome words from the given sentence. If there are no palindromes, print "No palindromes found".

Input Format

The input contains a single string S representing a sentence.

Output Format

The output prints all palindromic words separated by a space.

If no palindrome exists, print "No palindromes found".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: madam went to school

Output: madam

Answer

```
import java.util.Scanner;

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

        String[] words = sentence.split(" ");
        boolean found = false;

        for (String word : words) {
            if (isPalindrome(word)) {
                System.out.println(word);
                found = true;
            }
        }

        if (!found) {
            System.out.println("No palindromes found");
        }
    }

    static boolean isPalindrome(String word) {
        int left = 0, right = word.length() - 1;
        while (left < right) {
            if (word.charAt(left) != word.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return word.length() > 1; // Only consider words with length > 1
    }
}
```

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

4. Problem Statement

Sana is analyzing text for a secret code. She wants to find all words in a sentence that start and end with the same letter. These words are considered "special words" for her analysis.

Your task is to write a program that extracts and prints all words that start and end with the same letter (case-insensitive).

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

Input Format

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

Output Format

The output prints all words that start and end with the same letter separated by a space.

If no word satisfies the condition, print "No special words found".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Anna went to the civic center
Output: Anna civic

Answer

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

      String[] words = sentence.split(" ");
      boolean found = false;
```

```
        for (String word : words) {
            if (word.length() >= 1) {
                char first = Character.toLowerCase(word.charAt(0));
                char last = Character.toLowerCase(word.charAt(word.length() - 1));
                if (first == last) {
                    System.out.print(word + " ");
                    found = true;
                }
            }
        }
        if (!found) {
            System.out.println("No special words found");
        }
    }
}
```

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