Rajalakshmi Engineering College

Name: Sruthi Dinesh
Email: 240701536@rajalakshmi.edu.in
Roll no: 240701536
Phone: 7845725087
Branch: REC
Department: CSE - Section 7
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 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

```
import java.util.*;
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 (isValid(username)) {
            System.out.println("YES");
         } else {
            System.out.println("NO");
         }
      }

      sc.close();
   }

   public static boolean isValid(String s) {
```

```
        if (s.length() < 5 || s.length() > 15) return false;
        if (!Character.isLetter(s.charAt(0))) return false;

        for (int i = 0; i < s.length(); i++) {
            if (!Character.isLetterOrDigit(s.charAt(i))) return false;
        }

        return true;
    }
}
```

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

```
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(" ");
        boolean found = false;
        for (String word : words) {
            if (word.length() >= 5) {
                System.out.print(word + " ");
                found = true;
            }
        }
        if (!found) {
            System.out.print("No long words found");
        }
    }
}
```

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

3. 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;
class RepetitiveWords {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String sentence = sc.nextLine();
      String[] words = sentence.split(" ");
      String result = "";
      for (String word : words) {
         if (hasRepeatedChar(word)) {
            result += word + " ";
         }
      }
      if (result.equals("")) {
         System.out.println("No repetitive words found");
      } else {
         System.out.println(result.trim());
      }
   }

   private static boolean hasRepeatedChar(String word) {
      int[] freq = new int[256];
      for (int i = 0; i < word.length(); i++) {
         char c = word.charAt(i);
         freq[c]++;
         if (freq[c] > 1) return true;
```

```
        }
        return false;
    }
}
```

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 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;
class LongestWord {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String sentence = sc.nextLine();
        String[] words = sentence.split(" ");
```

```
        String longest = "";
        for (String word : words) {
            if (word.length() > longest.length()) {
                longest = word;
            }
        }
        System.out.println(longest);
    }
}
```

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