Rajalakshmi Engineering College

Name: Kaaviya Sri PS
Email: 240701222@rajalakshmi.edu.in
Roll no: 240701222
Phone: 8838174850
Branch: REC
Department: CSE - Section 6
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.*;

class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String[] words = sc.nextLine().split("\\s+");
      StringBuilder result = new StringBuilder();
      for (String word : words) {
         if (hasRepeatedChars(word)) {
            if (result.length() > 0) result.append(" ");
            result.append(word);
         }
      }
      if (result.length() > 0) System.out.println(result);
      else System.out.println("No repetitive words found");
   }

   static boolean hasRepeatedChars(String word) {
      boolean[] chars = new boolean[256];
      for (char c : word.toCharArray()) {
         if (chars[c]) return true;
         chars[c] = true;
      }
      return false;
   }
}
```

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

2. 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.*;
```

```
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");
            }
        }
    }

    static boolean isValid(String username) {
        if (username.length() < 5 || username.length() > 15) return false;
        char first = username.charAt(0);
        if (!Character.isLetter(first)) return false;
        for (char c : username.toCharArray()) {
            if (!Character.isLetterOrDigit(c)) return false;
        }
        return true;
    }
}
```

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

3. 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.*;

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();
         String[] words = title.split("\\s+");
         int count = 0;
         for (String word : words) {
            if (!word.isEmpty() && Character.isUpperCase(word.charAt(0))) {
               count++;
            }
         }
         System.out.println(count);
      }
   }
}
```

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

4. 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

```
import java.util.*;

class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      String sentence = scanner.nextLine();
      String[] words = sentence.split(" ");
      boolean found = false;

      for (String word : words) {
         if (word.matches(".*\\d.*")) {
            System.out.print(word + " ");
            found = true;
            }
        }

        if (!found) {
            System.out.println("No words with digits found");
        }
    }
}
```

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