Rajalakshmi Engineering College

Name: Dharshini A
Email: 241001049@rajalakshmi.edu.in
Roll no: 241001049
Phone: 8056618801
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a publishing company, editors often need to quickly analyze passages of text to check for punctuation usage. To assist them, you are asked to write a program that counts the number of specific punctuation marks in each passage.

The punctuation marks of interest are:

Commas (,)Periods (.)Question marks (?)

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 commas, periods, and question marks 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. How are you?
Output: 1 1 1

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 line = sc.nextLine();
         int commas = 0, periods = 0, questions = 0;
         for (char c : line.toCharArray()) {
            if (c == ',') commas++;
            else if (c == '.') periods++;
            else if (c == '?') questions++;
         }
         System.out.println(commas + " " + periods + " " + questions);
      }
   }
}
```

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

Rajalakshmi Engineering College

Name: Dharshini A
Email: 241001049@rajalakshmi.edu.in
Roll no: 241001049
Phone: 8056618801
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Anu is developing a tool for a conference registration system. Participants submit keywords related to their fields of interest. The organizer wants to sort these keywords alphabetically to generate tags for session grouping.

Write a program that accepts at least five keywords as input arguments and outputs them in sorted alphabetical order.

Input Format

The first line of input contains an integer n, representing the number of keywords.

The second line of input contains n space-separated keywords (string).

Output Format

The output prints n space separated strings representing the sorted keyword in alphabetical order.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5
Blockchain Cloud AI Data Cybersecurity
Output: AI Blockchain Cloud Cybersecurity Data

Answer

```
import java.util.*;

class KeywordSorter {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        String[] keywords = new String[n];
        for (int i = 0; i < n; i++) {
            keywords[i] = sc.next();
        }
        Arrays.sort(keywords);
        for (int i = 0; i < n; i++) {
            System.out.print(keywords[i]);
            if (i < n - 1) System.out.print(" ");
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: Dharshini A
Email: 241001049@rajalakshmi.edu.in
Roll no: 241001049
Phone: 8056618801
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Bechan Chacha is seeking help to filter out valid mobile numbers from a list provided by his crush. He can only pick his crush's number if the list contains valid mobile numbers.

A mobile number is considered valid if:

It has exactly 10 digits.It consists only of numeric values (0–9).It does not begin with zero.

Your task is to determine whether each mobile number in the list is valid or not.

Input Format

The first line contains an integer T, representing the number of mobile numbers

to check.

The next T lines each contain a string S, representing a mobile number.

Output Format

For each mobile number S, the output print "YES" if it is valid.

Otherwise, print "NO".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1
9876543210
Output: YES

Answer

```
import java.util.*;

class MobileNumberValidator {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int T = sc.nextInt();
        for (int i = 0; i < T; i++) {
            String S = sc.next();
            if (S.length() == 10 && S.matches("[1-9][0-9]{9}")) {
                System.out.println("YES");
            } else {
                System.out.println("NO");
            }
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: Dharshini A
Email: 241001049@rajalakshmi.edu.in
Roll no: 241001049
Phone: 8056618801
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q4

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Arjun is learning how to filter words from a sentence based on grammar rules. He wants to identify the valid words in a sentence.

A word is considered valid if it satisfies all these conditions:

The word contains only alphabets (a–z, A–Z).The word length is at least 2 characters.The word should not contain digits or special characters.

Your task is to read a sentence and print all the valid words in it.

Input Format

The input contains a single line containing a sentence S.

Output Format

The output prints all the valid words separated by spaces.

If no valid word exists, print "No valid words."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Hello world1 123 ab" @#$ Hi
Output: Hello Hi

Answer

```
import java.util.Scanner;

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String sentence = sc.nextLine();
      String[] words = sentence.split("\\s+");
      StringBuilder result = new StringBuilder();

      for (String word : words) {
         if (word.matches("[a-zA-Z]{2,}")) {
            result.append(word).append(" ");
         }
      }

      if (result.length() > 0) {
         System.out.println(result.toString().trim());
      } else {
         System.out.println("No valid words.");
      }
   }
}
```

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

Rajalakshmi Engineering College

Name: Dharshini A
Email: 241001049@rajalakshmi.edu.in
Roll no: 241001049
Phone: 8056618801
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 4_Q5

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

In a secure banking system, customers are required to create PIN codes for accessing their accounts. The bank wants to validate these PIN codes before accepting them.

A PIN code is considered valid if:

It consists of exactly 4 digits.All characters must be numeric (0–9).It cannot contain all identical digits (e.g., 1111 is invalid).

Your task is to determine whether each PIN code in the list is valid or not.

Input Format

The first line of input contains an integer T, representing the number of PIN codes to check.

The next T lines each contain a string S, representing a PIN code.

Output Format

For each PIN code 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
1234
Output: YES

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 pin = sc.nextLine();
            if (pin.matches("\\d{4}") && !pin.chars().allMatch(ch -> ch ==
pin.charAt(0))) {
                System.out.println("YES");
            } else {
                System.out.println("NO");
            }
        }
    }
}
```

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

Rajalakshmi Engineering College

Name: Dharshini A
Email: 241001049@rajalakshmi.edu.in
Roll no: 241001049
Phone: 8056618801
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

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

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;

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 c : passage.toCharArray()) {
                if ("aeiouAEIOU".indexOf(c) != -1) {
                    count++;
                }
            }
            System.out.println(count);
        }
    }
}
```

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

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

```
    }
}
```

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

3. 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;

class Main {
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
        String[] words = sc.nextLine().split("\\s+");
        StringBuilder sb = new StringBuilder();
        for (String word : words) {
            if (word.length() > 0) {
                char start = Character.toLowerCase(word.charAt(0));
                char end = Character.toLowerCase(word.charAt(word.length() - 1));
                if (start == end) {
                    sb.append(word).append(" ");
                }
            }
        }
        if (sb.length() > 0) {
            System.out.println(sb.toString().trim());
        } else {
            System.out.println("No special words found");
        }
    }
}
```

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

4. 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;

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

    static boolean isPalindrome(String s) {
        int left = 0, right = s.length() - 1;
        while (left < right) {
            if (s.charAt(left) != s.charAt(right)) return false;
            left++;
            right--;
        }
        return true;
    }
}
```

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

Rajalakshmi Engineering College

Name: Dharshini A
Email: 241001049@rajalakshmi.edu.in
Roll no: 241001049
Phone: 8056618801
Branch: REC
Department: IT - Section 2
Batch: 2028
Degree: B.E - IT

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;

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[] words = sc.nextLine().split("\\s+");
        StringBuilder sb = new StringBuilder();

        for (String word : words) {
            if (hasRepeatedChars(word)) {
                sb.append(word).append(" ");
            }
        }
        if (sb.length() > 0) {
            System.out.println(sb.toString().trim());
        } else {
            System.out.println("No repetitive words found");
        }
    }

    static boolean hasRepeatedChars(String word) {
        HashSet<Character> set = new HashSet<>();
        for (char c : word.toCharArray()) {
            if (set.contains(c)) return true;
            set.add(c);
        }
```

```
        return false;
    }
}
```

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

2. 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

```
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
        String[] words = sc.nextLine().split("\\s+");
        StringBuilder sb = new StringBuilder();
        for (String word : words) {
            if (countVowels(word) >= 2) {
                sb.append(word).append(" ");
            }
        }
        if (sb.length() > 0) {
            System.out.println(sb.toString().trim());
        } else {
            System.out.println("No words with two vowels");
        }
    }

    static int countVowels(String word) {
        int count = 0;
        for (char c : word.toLowerCase().toCharArray()) {
            if ("aeiou".indexOf(c) != -1) count++;
        }
        return count;
    }
}
```

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

3. 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.Scanner;

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 (username.matches("^[a-zA-Z][a-zA-Z0-9]{4,14}$")) {
                System.out.println("YES");
            } else {
                System.out.println("NO");
            }
        }
    }
}
```

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 Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // Read the entire line as the book title
        String title = sc.nextLine();
        // Split the title into words using whitespace as delimiter
        String[] words = title.split("\\s+");

        String longestWord = "";
        // Iterate over each word to find the longest one
        for (String word : words) {
            if (word.length() > longestWord.length()) {
```

```
                longestWord = word;
            }
        }
        // Print the longest word found
        System.out.println(longestWord);
    }
}
```

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