

Q2:

PROGRAM : import java.util.Scanner;

```
public class UniqueCharactersFinder {

    public static int findLength(String text) {
        int length = 0;
        try {
            while (true) {
                text.charAt(length);
                length++;
            }
        } catch (StringIndexOutOfBoundsException e) {
            // Exception occurs when we reach beyond the string length
        }
        return length;
    }

    public static char[] findUniqueCharacters(String text) {
        int length = findLength(text);
        char[] allUniqueChars = new char[length]; // Temporary array to store unique chars
        int uniqueCount = 0;

        for (int i = 0; i < length; i++) {
            char currentChar = text.charAt(i);
            boolean isUnique = true;

            for (int j = 0; j < i; j++) {
                if (text.charAt(j) == currentChar) {
                    isUnique = false;
                    break;
                }
            }

            if (isUnique) {
                allUniqueChars[uniqueCount] = currentChar;
                uniqueCount++;
            }
        }

        char[] result = new char[uniqueCount];
```

```

        for (int i = 0; i < uniqueCount; i++) {
            result[i] = allUniqueChars[i];
        }

        return result;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        int length = findLength(input);
        System.out.println("Length of string (custom method): " + length);

        char[] uniqueChars = findUniqueCharacters(input);

        System.out.println("Original string: " + input);
        System.out.print("Unique characters: ");
        for (char c : uniqueChars) {
            System.out.print(c + " ");
        }
        System.out.println("\nTotal unique characters: " + uniqueChars.length);

        scanner.close();
    }
}

```

OUTPUT:

```

Enter a string: hello
Length of string (custom method): 5
Original string: hello
Unique characters: h e l o
Total unique characters: 4

```

Q3:

PROGRAM:

```

import java.util.Scanner;

```

```

public class FirstNonRepeatingCharacter {

    // Method to find the first non-repeating character in a string
    public static char findFirstNonRepeatingChar(String text) {
        // Create an array to store frequency of characters (256 ASCII characters)
        int[] frequency = new int[256];
        int length = text.length();

        // Step 1: Count frequency of each character
        for (int i = 0; i < length; i++) {
            char currentChar = text.charAt(i);
            frequency[currentChar]++;
        }

        // Step 2: Find the first character with frequency 1
        for (int i = 0; i < length; i++) {
            char currentChar = text.charAt(i);
            if (frequency[currentChar] == 1) {
                return currentChar;
            }
        }

        // If no non-repeating character found, return a special character
        return '\0'; // null character
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Find the first non-repeating character
        char result = findFirstNonRepeatingChar(input);

        // Display results
        System.out.println("Original string: " + input);

        if (result != '\0') {
            System.out.println("First non-repeating character: " + result);
        } else {
            System.out.println("No non-repeating character found in the string");
        }
    }
}

```

```
        scanner.close();
    }
}
```

OUTPUT:

Enter a string: programming

Original string: programming

First non-repeating character: p

Q4:

PROGRAM:

```
import java.util.Scanner;
```

```
public class CharacterFrequency {
```

```
    // Method to find the frequency of characters in a string
```

```
    public static int[][] findCharacterFrequency(String text) {
```

```
        // Create an array to store frequency of characters (256 ASCII characters)
```

```
        int[] frequency = new int[256];
```

```
        int length = text.length();
```

```
        // Step 1: Count frequency of each character
```

```
        for (int i = 0; i < length; i++) {
```

```
            char currentChar = text.charAt(i);
```

```
            frequency[currentChar]++;
```

```
        }
```

```
        // Step 2: Count how many unique characters we have
```

```
        int uniqueCount = 0;
```

```
        for (int i = 0; i < 256; i++) {
```

```
            if (frequency[i] > 0) {
```

```
                uniqueCount++;
```

```
            }
```

```
        }
```

```
        // Step 3: Create a 2D array to store characters and their frequencies
```

```
        int[][] result = new int[uniqueCount][2];
```

```
        int index = 0;
```

```
        for (int i = 0; i < 256; i++) {
```

```

        if (frequency[i] > 0) {
            result[index][0] = i;    // Store ASCII value (character)
            result[index][1] = frequency[i]; // Store frequency
            index++;
        }
    }

    return result;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter a string: ");
    String input = scanner.nextLine();

    // Find character frequencies
    int[][] frequencyData = findCharacterFrequency(input);

    // Display results
    System.out.println("Original string: " + input);
    System.out.println("Character frequencies:");
    System.out.println("-----");

    for (int i = 0; i < frequencyData.length; i++) {
        char character = (char) frequencyData[i][0];
        int frequency = frequencyData[i][1];
        System.out.println("'" + character + "' : " + frequency + " time(s)");
    }

    System.out.println("Total unique characters: " + frequencyData.length);

    scanner.close();
}
}

```

#### OUTPUT:

```

Enter a string: hello
Original string: hello
Character frequencies:
-----
'h' : 1 time(s)
'e' : 1 time(s)
'l' : 2 time(s)

```

'o' : 1 time(s)

Total unique characters: 4

Q5:

PROGRAM:

```
import java.util.Scanner;
```

```
public class CharacterFrequencyWithUnique {
```

```
    // Method to find unique characters in a string using nested loops
```

```
    public static char[] findUniqueCharacters(String text) {
```

```
        int length = text.length();
```

```
        char[] tempUniqueChars = new char[length];
```

```
        int uniqueCount = 0;
```

```
        // Use nested loops to find unique characters
```

```
        for (int i = 0; i < length; i++) {
```

```
            char currentChar = text.charAt(i);
```

```
            boolean isUnique = true;
```

```
            // Check if this character has appeared before
```

```
            for (int j = 0; j < i; j++) {
```

```
                if (text.charAt(j) == currentChar) {
```

```
                    isUnique = false;
```

```
                    break;
```

```
                }
```

```
            }
```

```
            // If unique, add to temporary array
```

```
            if (isUnique) {
```

```
                tempUniqueChars[uniqueCount] = currentChar;
```

```
                uniqueCount++;
```

```
            }
```

```
        }
```

```
        // Create final array with exact size
```

```
        char[] uniqueChars = new char[uniqueCount];
```

```
        for (int i = 0; i < uniqueCount; i++) {
```

```
            uniqueChars[i] = tempUniqueChars[i];
```

```
        }
```

```
        return uniqueChars;
```

```

}

// Method to find frequency of characters using unique characters
public static String[][] findCharacterFrequency(String text) {
    // Get unique characters first
    char[] uniqueChars = findUniqueCharacters(text);
    int length = text.length();

    // Create frequency array for ASCII characters
    int[] frequency = new int[256];

    // Count frequency of each character using ASCII indexing
    for (int i = 0; i < length; i++) {
        char currentChar = text.charAt(i);
        frequency[currentChar]++;
    }

    // Create 2D String array to store character-frequency pairs
    String[][] result = new String[uniqueChars.length][2];

    // Fill the 2D array with unique characters and their frequencies
    for (int i = 0; i < uniqueChars.length; i++) {
        char currentChar = uniqueChars[i];
        result[i][0] = String.valueOf(currentChar); // Character as string
        result[i][1] = String.valueOf(frequency[currentChar]); // Frequency as string
    }

    return result;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter a string: ");
    String input = scanner.nextLine();

    // Find unique characters
    char[] uniqueChars = findUniqueCharacters(input);

    // Find character frequencies
    String[][] frequencyData = findCharacterFrequency(input);

    // Display results
    System.out.println("\nOriginal string: " + input);
}

```

```

        System.out.print("Unique characters found: ");
        for (char c : uniqueChars) {
            System.out.print(c + " ");
        }
        System.out.println("\n");

        System.out.println("Character frequencies:");
        System.out.println("-----");

        for (int i = 0; i < frequencyData.length; i++) {
            String character = frequencyData[i][0];
            String frequency = frequencyData[i][1];
            System.out.println("'" + character + "' : " + frequency + " time(s)");
        }

        System.out.println("Total unique characters: " + uniqueChars.length);
        System.out.println("Total characters in string: " + input.length());

        scanner.close();
    }
}

```

OUTPUT:

Enter a string: hello

Original string: hello

Unique characters found: h e l o

Character frequencies:

-----

'h' : 1 time(s)

'e' : 1 time(s)

'l' : 2 time(s)

'o' : 1 time(s)

Total unique characters: 4

Total characters in string: 5

Q6:

PROGRAM :



```

import java.util.Scanner;

public class CharacterFrequencyNestedLoops {

    // Method to find frequency of characters using nested loops
    public static String[] findCharacterFrequency(String text) {
        // Convert string to character array
        char[] charArray = text.toCharArray();
        int length = charArray.length;

        // Create arrays to store frequencies and track processed characters
        int[] frequency = new int[length];
        char[] characters = charArray.clone();

        // Initialize all frequencies to 1
        for (int i = 0; i < length; i++) {
            frequency[i] = 1;
        }

        // Use nested loops to find frequencies
        for (int i = 0; i < length; i++) {
            // Skip if character is already marked as processed
            if (characters[i] == '0') {
                continue;
            }

            for (int j = i + 1; j < length; j++) {
                // If duplicate character found
                if (characters[i] == characters[j]) {
                    frequency[i]++; // Increment frequency
                    characters[j] = '0'; // Mark as processed to avoid recounting
                }
            }
        }

        // Count how many unique characters we have (non-zero characters)
        int uniqueCount = 0;
        for (int i = 0; i < length; i++) {
            if (characters[i] != '0') {
                uniqueCount++;
            }
        }

        // Create 1D String array to store character-frequency pairs
    }
}

```

```

String[] result = new String[uniqueCount];
int index = 0;

for (int i = 0; i < length; i++) {
    if (characters[i] != '0') {
        result[index] = "" + characters[i] + " : " + frequency[i] + " time(s)";
        index++;
    }
}

return result;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter a string: ");
    String input = scanner.nextLine();

    // Find character frequencies using nested loops
    String[] frequencyData = findCharacterFrequency(input);

    // Display results
    System.out.println("\nOriginal string: " + input);
    System.out.println("Character frequencies (using nested loops):");
    System.out.println("-----");

    for (String data : frequencyData) {
        System.out.println(data);
    }

    System.out.println("Total unique characters: " + frequencyData.length);
    System.out.println("Total characters in string: " + input.length());

    scanner.close();
}
}

```

OUTPUT:

Enter a string: java

Original string: java

Character frequencies (using nested loops):

-----  
'j' : 1 time(s)  
'a' : 2 time(s)  
'v' : 1 time(s)  
Total unique characters: 3  
Total characters in string: 4

Q8:

PROGRAM:

```
import java.util.Scanner;
```

```
public class AnagramChecker {
```

```
    // Method to check if two texts are anagrams
```

```
    public static boolean areAnagrams(String text1, String text2) {
```

```
        // Step 1: Check if lengths are equal
```

```
        if (text1.length() != text2.length()) {
```

```
            return false;
```

```
        }
```

```
        // Step 2: Create arrays to store frequency of characters (256 ASCII characters)
```

```
        int[] freq1 = new int[256];
```

```
        int[] freq2 = new int[256];
```

```
        // Step 3: Find frequency of characters in first text
```

```
        for (int i = 0; i < text1.length(); i++) {
```

```
            char c = text1.charAt(i);
```

```
            freq1[c]++;
```

```
        }
```

```
        // Step 3: Find frequency of characters in second text
```

```
        for (int i = 0; i < text2.length(); i++) {
```

```
            char c = text2.charAt(i);
```

```
            freq2[c]++;
```

```
        }
```

```
        // Step 4: Compare frequency of characters
```

```
        for (int i = 0; i < 256; i++) {
```

```
            if (freq1[i] != freq2[i]) {
```

```
                return false;
```

```

    }
}

return true;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter first text: ");
    String text1 = scanner.nextLine();

    System.out.print("Enter second text: ");
    String text2 = scanner.nextLine();

    // Check if texts are anagrams
    boolean result = areAnagrams(text1, text2);

    // Display results
    System.out.println("\nText 1: " + text1);
    System.out.println("Text 2: " + text2);

    if (result) {
        System.out.println("✓ These texts are ANAGRAMS!");
    } else {
        System.out.println("✗ These texts are NOT anagrams.");
    }

    scanner.close();
}
}

```

OUTPUT:

Enter first text: listen  
Enter second text: silent

Text 1: listen  
Text 2: silent  
These texts are ANAGRAMS!