**1. An organization took up the exercise to find the Body Mass Index (BMI) of all the persons in**

**a team of 10 members. For this create a program to find the BMI and display the height,**

**weight, BMI, and status of each individual**

**Hint =>**

**a. Take user input for the person's weight (kg) and height (cm) and store it in the**

**corresponding 2D array of 10 rows. The First Column stores the weight and the second**

**column stores the height in cm**

**b. Create a Method to find the BMI and status of every person given the person's height**

**and weight and return the 2D String array. Use the formula BMI = weight / (height \***

**height). Note unit is kg/m^2. For this convert cm to meter**

**c. Create a Method that takes the 2D array of height and weight as parameters. Calls the**

**user-defined method to compute the BMI and the BMI Status and stores in a 2D String**

**array of height, weight, BMI, and status.**

**d. Create a method to display the 2D string array in a tabular format of Person's Height,**

**Weight, BMI, and the Status**

**e. Finally, the main function takes user inputs, calls the user-defined methods, and displays**

**the result.**

import java.util.Scanner;

public class BMICalculator {

// Method to calculate BMI and status

public static String[] calculateBMI(double weight, double heightCm) {

double heightM = heightCm / 100.0; // convert cm to meters

double bmi = weight / (heightM \* heightM);

String status;

if (bmi < 18.5) {

status = "Underweight";

} else if (bmi < 25) {

status = "Normal";

} else if (bmi < 30) {

status = "Overweight";

} else {

status = "Obese";

}

return new String[]{

String.valueOf(heightCm),

String.valueOf(weight),

String.format("%.2f", bmi), // 2 decimal places

status

};

}

// Method to compute BMI table

public static String[][] computeBMITable(double[][] data) {

String[][] result = new String[data.length][4];

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

double weight = data[i][0];

double height = data[i][1];

result[i] = calculateBMI(weight, height);

}

return result;

}

// Method to display table

public static void displayTable(String[][] table) {

System.out.printf("%-10s %-10s %-10s %-15s\n", "Height(cm)", "Weight(kg)", "BMI", "Status");

System.out.println("-----------------------------------------------------");

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

System.out.printf("%-10s %-10s %-10s %-15s\n",

table[i][0], table[i][1], table[i][2], table[i][3]);

}

}

// MAIN

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

double[][] data = new double[10][2]; // 10 rows, 2 cols (weight, height)

// Take input

for (int i = 0; i < 10; i++) {

System.out.println("Enter details for Person " + (i + 1));

System.out.print("Weight (kg): ");

data[i][0] = sc.nextDouble();

System.out.print("Height (cm): ");

data[i][1] = sc.nextDouble();

}

// Compute BMI Table

String[][] bmiTable = computeBMITable(data);

// Display Table

System.out.println("\n=== BMI REPORT ===");

displayTable(bmiTable);

sc.close();

}

}

**2. Find unique characters in a string using the charAt() method and display the result**

**Hint =>**

**a. Create a Method to find the length of the text without using the String method length()**

**b. Create a method to Find unique characters in a string using the charAt() method and**

**return them as a 1D array. The logic used here is as follows:**

**i. Create an array to store the unique characters in the text. The size is the length of**

**the text**

**ii. Loops to Find the unique characters in the text. Find the unique characters in the text**

**using a nested loop. An outer loop iterates through each character and an inner loop**

**checks if the character is unique by comparing it with the previous characters. If the**

**character is unique, it is stored in the result array**

**iii. Create a new array to store the unique characters**

**c. Finally, the main function takes user inputs, calls the user-defined methods, and displays**

**the result.**

import java.util.Scanner;

public class UniqueCharacters {

// Method to find length without using length()

public static int findLength(String str) {

int count = 0;

try {

while (true) {

str.charAt(count); // throws exception when out of bounds

count++;

}

} catch (StringIndexOutOfBoundsException e) {

// stop

}

return count;

}

// Method to find unique characters

public static char[] findUniqueChars(String str) {

int len = findLength(str);

char[] temp = new char[len]; // temp array for unique chars

int uniqueCount = 0;

for (int i = 0; i < len; i++) {

char current = str.charAt(i);

boolean isUnique = true;

// check if already exists in temp

for (int j = 0; j < uniqueCount; j++) {

if (temp[j] == current) {

isUnique = false;

break;

}

}

if (isUnique) {

temp[uniqueCount++] = current;

}

}

// copy only unique chars into new array

char[] uniqueChars = new char[uniqueCount];

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

uniqueChars[i] = temp[i];

}

return uniqueChars;

}

// MAIN

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Input

System.out.print("Enter a string: ");

String text = sc.nextLine();

// Find unique characters

char[] unique = findUniqueChars(text);

// Display result

System.out.println("\n--- Unique Characters ---");

for (char c : unique) {

System.out.print(c + " ");

}

System.out.println();

sc.close();

}

}

**3. Write a program to find the first non-repeating character in a string and show the result**

**Hint =>**

**a. Non-repeating character is a character that occurs only once in the string**

**b. Create a Method to find the first non-repeating character in a string using the charAt()**

**method and return the character. The logic used here is as follows:**

**i. Create an array to store the frequency of characters in the text. ASCII values of**

**characters are used as indexes in the array to store the frequency of each character.**

**There are 256 ASCII characters**

**ii. Loop through the text to find the frequency of characters in the text**

**iii. Loop through the text to find the first non-repeating character in the text by checking**

**the frequency of each character**

**c. In the main function take user inputs, call user-defined methods, and displays result.**

import java.util.Scanner;

public class FirstNonRepeatingChar {

// Method to find first non-repeating character

public static char findFirstNonRepeating(String str) {

int[] freq = new int[256]; // ASCII frequency array

// Step 1: Count frequencies

try {

int i = 0;

while (true) {

char ch = str.charAt(i);

freq[ch]++; // increment frequency

i++;

}

} catch (StringIndexOutOfBoundsException e) {

// finished counting

}

// Step 2: Find first char with frequency 1

try {

int i = 0;

while (true) {

char ch = str.charAt(i);

if (freq[ch] == 1) {

return ch;

}

i++;

}

} catch (StringIndexOutOfBoundsException e) {

// finished searching

}

return '\0'; // null char if no non-repeating character

}

// MAIN

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Input

System.out.print("Enter a string: ");

String text = sc.nextLine();

// Find first non-repeating character

char result = findFirstNonRepeating(text);

// Display

if (result != '\0') {

System.out.println("First non-repeating character: " + result);

} else {

System.out.println("No non-repeating character found.");

}

sc.close();

}

}

**4. Write a program to find the frequency of characters in a string using the charAt() method and**

**display the result**

**Hint =>**

**a. Create a method to find the frequency of characters in a string using the charAt() method**

**and return the characters and their frequencies in a 2D array. The logic used here is as**

**follows:**

**i. Create an array to store the frequency of characters in the text. ASCII values of**

**characters are used as indexes in the array to store the frequency of each character.**

**There are 256 ASCII characters**

**ii. Loop through the text to find the frequency of characters in the text**

**iii. Create an array to store the characters and their frequencies**

**iv. Loop through the characters in the text and store the characters and their**

**frequencies**

**b. In the main function take user inputs, call user-defined methods, and displays result.**

import java.util.Scanner;

public class CharFrequency {

// Method to calculate frequency and return 2D array

public static String[][] getCharFrequency(String str) {

int[] freq = new int[256]; // ASCII frequency array

// Step 1: Count frequency of each character

try {

int i = 0;

while (true) {

char ch = str.charAt(i);

freq[ch]++;

i++;

}

} catch (StringIndexOutOfBoundsException e) {

// done counting

}

// Step 2: Count how many unique characters

int uniqueCount = 0;

try {

int i = 0;

while (true) {

char ch = str.charAt(i);

if (freq[ch] > 0) {

uniqueCount++;

freq[ch] = freq[ch]; // just ensure it's counted

}

i++;

}

} catch (StringIndexOutOfBoundsException e) {

// done

}

// Step 3: Create result array (unique characters only)

String[][] result = new String[uniqueCount][2];

int index = 0;

boolean[] added = new boolean[256]; // to avoid duplicates

try {

int i = 0;

while (true) {

char ch = str.charAt(i);

if (!added[ch]) {

result[index][0] = String.valueOf(ch);

result[index][1] = String.valueOf(freq[ch]);

added[ch] = true;

index++;

}

i++;

}

} catch (StringIndexOutOfBoundsException e) {

// finished storing

}

return result;

}

// Method to display 2D array in tabular format

public static void displayResult(String[][] arr) {

System.out.println("\nCharacter | Frequency");

System.out.println("----------------------");

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

System.out.println(" " + arr[i][0] + " | " + Integer.parseInt(arr[i][1]));

}

}

// MAIN

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Input

System.out.print("Enter a string: ");

String text = sc.nextLine();

// Get frequencies

String[][] result = getCharFrequency(text);

// Display

displayResult(result);

sc.close();

}

}

**5. Write a program to find the frequency of characters in a string using unique characters and**

**display the result**

**Hint =>**

**a. Create a method to Find unique characters in a string using the charAt() method and**

**return them as a 1D array. Use Nested Loops to find the unique characters in the text**

**b. Create a method to find the frequency of characters in a string and return the characters**

**and their frequencies in a 2D array. The logic used here is as follows:**

**i. Create an array to store the frequency of characters in the text. ASCII values of**

**characters are used as indexes in the array to store the frequency of each character.**

**There are 256 ASCII characters**

**ii. Loop through the text to find the frequency of characters in the text**

**iii. Call the uniqueCharacters() method to find the unique characters in the text**

**iv. Create a 2D String array to store the unique characters and their frequencies.**

**v. Loop through the unique characters and store the characters and their frequencies**

import java.util.Scanner;

public class CharFrequencyWithUnique {

// Method to find length of string (without length())

public static int getLength(String str) {

int count = 0;

try {

while (true) {

str.charAt(count);

count++;

}

} catch (StringIndexOutOfBoundsException e) {

// reached end

}

return count;

}

// Method to find unique characters

public static char[] uniqueCharacters(String str) {

int n = getLength(str);

char[] temp = new char[n];

int uniqueCount = 0;

for (int i = 0; i < n; i++) {

char current = str.charAt(i);

boolean isUnique = true;

// check if already added

for (int j = 0; j < uniqueCount; j++) {

if (temp[j] == current) {

isUnique = false;

break;

}

}

if (isUnique) {

temp[uniqueCount] = current;

uniqueCount++;

}

}

// store only uniqueCount chars

char[] uniqueChars = new char[uniqueCount];

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

uniqueChars[i] = temp[i];

}

return uniqueChars;

}

// Method to find frequency of characters (using unique characters)

public static String[][] getCharFrequency(String str) {

int[] freq = new int[256]; // ASCII-based frequency array

int n = getLength(str);

// count frequency

for (int i = 0; i < n; i++) {

freq[str.charAt(i)]++;

}

// get unique characters

char[] unique = uniqueCharacters(str);

// prepare result 2D array

String[][] result = new String[unique.length][2];

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

result[i][0] = String.valueOf(unique[i]);

result[i][1] = String.valueOf(freq[unique[i]]);

}

return result;

}

// Method to display 2D array

public static void displayResult(String[][] arr) {

System.out.println("\nCharacter | Frequency");

System.out.println("----------------------");

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

System.out.println(" " + arr[i][0] + " | " + Integer.parseInt(arr[i][1]));

}

}

// MAIN

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// input

System.out.print("Enter a string: ");

String text = sc.nextLine();

// get frequency

String[][] result = getCharFrequency(text);

// display

displayResult(result);

sc.close();

}

}

**6. Write a program to find the frequency of characters in a string using nested loops and**

**display the result**

**Hint =>**

**a. Create a method to find the frequency of characters in a string and return the characters**

**and their frequencies in a 1D array. The logic used here is as follows:**

**i. Create an array to store the frequency of each character in the text and an array to**

**store the characters in the text using the toCharArray() method**

**ii. Loops to Find the frequency of each character in the text and store the result in a**

**frequency array. For this use a Nested Loop with an Outer loop to iterate through**

**each character in the text and initialize the frequency of each character to 1. And an**

**Inner loop to check for duplicate characters. In case of duplicate increment the**

**frequency value and set the duplicate characters to '0' to avoid counting them again.**

**iii. Create a 1D String array to store the characters and their frequencies. For this**

**Iterate through the characters in the text and store the characters and their**

**frequencies**

**b. Finally, the main function takes user inputs, calls the user-defined methods, and displays**

**the result.**

import java.util.Scanner;

public class CharFrequencyNestedLoops {

// Method to find frequency using nested loops

public static String[] findFrequency(String str) {

char[] chars = str.toCharArray(); // convert string to char array

int n = chars.length;

int[] freq = new int[n];

// Initialize frequencies

for (int i = 0; i < n; i++) {

freq[i] = 1; // each character at least once

}

// Count frequencies using nested loop

for (int i = 0; i < n; i++) {

if (chars[i] == '0') continue; // skip already counted chars

for (int j = i + 1; j < n; j++) {

if (chars[i] == chars[j]) {

freq[i]++;

chars[j] = '0'; // mark as counted

}

}

}

// Create result array of unique characters

int uniqueCount = 0;

for (int i = 0; i < n; i++) {

if (chars[i] != '0') uniqueCount++;

}

String[] result = new String[uniqueCount];

int idx = 0;

for (int i = 0; i < n; i++) {

if (chars[i] != '0') {

result[idx] = chars[i] + " : " + freq[i];

idx++;

}

}

return result;

}

// Method to display result

public static void displayResult(String[] arr) {

System.out.println("\nCharacter | Frequency");

System.out.println("----------------------");

for (String s : arr) {

String[] parts = s.split(" : ");

System.out.println(" " + parts[0] + " | " + parts[1]);

}

}

// MAIN

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Input

System.out.print("Enter a string: ");

String text = sc.nextLine();

// Get frequencies

String[] result = findFrequency(text);

// Display

displayResult(result);

sc.close();

}

}