**Program Title:** Cafeteria Billing System

Create a Java program that simulates a basic cafeteria billing system. The program should allow customers to select items from a predefined menu, specify the quantity, and calculate the total bill based on their selections.

**Menu:**

1. Coffee - Rs 50

2. Sandwich - Rs 70

3. Burger - Rs 100

0. Exit

**Task:**

1. Display the menu with item numbers and prices.
2. Prompt the user to enter the item number repeatedly until they choose to exit by entering 0.
3. For each valid item selection (1 to 3):

* Ask the user for the quantity.
* Calculate and display the **subtotal** for that item (price × quantity).

1. After the user exits, display the **total bill amount** formatted to two decimal places.
2. Use a separate method calculateSubtotal(int itemNumber, int quantity) to compute the subtotal for each item.

**Sample Input:**

1 (Enter item number (0 to finish))

2 (Enter quantity)

Subtotal for Coffee (x2): Rs100.0

3 ((Enter item number (0 to finish))

1 (Enter quantity)

Subtotal for Burger (x1): Rs100.0

0 (Enter item number (0 to finish))

Total Bill: Rs200.00

**Sample Output:**

1.Coffee - Rs 50

2.Sandwich - Rs 70

3.Burger - Rs 100

0.Exit

Subtotal for Coffee (x2): Rs100.0

Subtotal for Burger (x1): Rs100.0

Total Bill: Rs200.00

**Program Title: Sum of Unique Elements in an Array**

**Objective:**

Write a Java program that takes an array of integers as input and calculates the sum of all **unique elements** (i.e., elements that appear **only once** in the array).

Implement your logic inside a method sumOfUnique(int[] arr)

**Sample Input/Output:**

**Input:**

Enter size of array:

6

Enter elements:

1 2 3 2 4 1

**Output:**

Sum of unique elements: 7

**Program Title: Character Type Counter in a String**

**Objective:**

Write a Java program to read a string from the user and count the number of **vowels**, **consonants**, and **digits** present in it.

The program should:

1. Prompt the user to enter a string (which may contain letters, digits, spaces, or special characters).
2. Count:
   * **Vowels** – Characters a, e, i, o, u (both uppercase and lowercase).
   * **Consonants** – All other **alphabet letters** that are **not vowels**.
   * **Digits** – Numeric characters from 0 to 9.
3. Display the counts of vowels, consonants, and digits separately.

**Implementation Details:**

* Use the following methods:
  + countVowels(String input) – Returns the number of vowels in the string.
  + countConsonants(String input) – Returns the number of consonants.
  + countDigits(String input) – Returns the number of digit characters.
* The program should ignore spaces and special characters when counting vowels/consonants.

**Sample Input/Output:**

**Input:**

Enter a string:

Hello World 123

**Output:**

Number of vowels: 3

Number of constants: 7

Number of digits: 3

**Program Title: Password Strength Validator**

Write a Java program to validate the strength of a user-entered password based on predefined security rules.

**Problem Description:**

The program should:

1. Prompt the user to **enter a password**.
2. Check whether the password is **strong** or **weak** based on the following conditions.

**Password Strength Rules:**

A password is considered **strong** if:

* It contains **at least 8 characters**.
* It **does not contain** the word "password" (case insensitive).
* It contains **at least one digit (0-9)**.
* It contains **at least one special character** from the following list: @, %, #, &, $.

If all the above conditions are satisfied, the password is considered **strong**; otherwise, it is **weak**.

**Input 1:**

Enter your password:

P@ssw0rd123

**Output 1:**

Password is strong

**Input 2:**

Enter your password:

password123

**Output 2:**

Password is weak