**Strings-M1 questions:**

1. **Character Count:**

You are tasked with creating a program that reads a word from the user, validates it, and then counts the occurrences of each character in the word. The program should print the count of each character in the order they appear in the word.

**Requirements:**

1. The program should prompt the user to enter a word.
2. The word should be converted to lowercase.
3. The program should validate that the word contains only alphabetic characters. If the word is invalid, print an error message and terminate the program.
4. The program should count the occurrences of each character in the word.
5. The program should print the count of each character in the order they appear in the word.

**Example Input and Output:**

**Test Case 1:**

Enter a word

hello

**Output:**

1211

**Explanation:**

* ‘h’ appears 1 time
* ‘e’ appears 1 time
* ‘l’ appears 2 times
* ‘o’ appears 1 time

**Test Case 2:**

Enter a word

world

**Output:**

11111

**Explanation:**

* ‘w’ appears 1 time
* ‘o’ appears 1 time
* ‘r’ appears 1 time
* ‘l’ appears 1 time
* ‘d’ appears 1 time

**Test Case 3:**

Enter a word

123abc

**Output:**

123abc is an invalid word

**Boilerplate Code:**

**Java**

import java.util.HashMap;

import java.util.\*;

import java.util.Scanner;

public class UserInterface {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Fill the code here

}

}

1. **Question: PothysWordBattle**

You are tasked with creating a program that reads a sentence from the user, validates it, and then counts the unique vowels or consonants in each word based on the word’s length. The program should print the count of vowels for words with even lengths and the count of consonants for words with odd lengths.

**Requirements:**

1. The program should prompt the user to enter a sentence.
2. The sentence should be validated to contain only alphabetic characters and spaces. If the sentence is invalid, print an error message and terminate the program.
3. For each word in the sentence:
   * If the word length is even, count the unique vowels in the word.
   * If the word length is odd, count the unique consonants in the word.
4. The program should print the count of vowels or consonants for each word based on the above criteria.

**Example Input and Output:**

**Test Case 1:**

Enter the sentence

hello world

**Output:**

There is 1 vowel in the word hello

There are 3 consonants in the word world

**Explanation:**

* ‘hello’ has an odd length (5), so count consonants: ‘h’, ‘l’, ‘o’ (3 unique consonants)
* ‘world’ has an even length (5), so count vowels: ‘o’ (1 unique vowel)

**Test Case 2:**

Enter the sentence

java programming

**Output:**

There are 2 vowels in the word java

There are 6 consonants in the word programming

**Explanation:**

* ‘java’ has an even length (4), so count vowels: ‘a’ (2 unique vowels)
* ‘programming’ has an odd length (11), so count consonants: ‘p’, ‘r’, ‘g’, ‘m’, ‘n’ (6 unique consonants)

**Test Case 3:**

Enter the sentence

123abc

**Output:**

Enter a valid sentence, the sentence should contain only alphabets and spaces

1. **Question: SecurityKeyGeneration**

You are tasked with creating a program that reads a username from the user, validates it, and then processes it to generate a specific output. The program should follow these steps:

1. Prompt the user to enter a username.
2. Validate that the username is between 2 and 10 characters long and contains only alphabetic characters and spaces. If the username is invalid, print an error message and terminate the program.
3. Convert the username to lowercase.
4. Calculate the ASCII average character of the username.
5. Count the occurrences of each character in the username.
6. Print the character counts followed by the ASCII average character.

**Example Input and Output:**

**Test Case 1:**

Enter the username:

Alice

**Output:**

a1l1i1c1e1c

**Explanation:**

* ‘Alice’ converted to lowercase is ‘alice’.
* Character counts: ‘a’ appears 1 time, ‘l’ appears 1 time, ‘i’ appears 1 time, ‘c’ appears 1 time, ‘e’ appears 1 time.
* ASCII average character of ‘alice’ is ‘c’.

**Test Case 2:**

Enter the username:

Bob

**Output:**

Invalid Username

**Explanation:**

* ‘Bob’ is valid, but the code has a bug in the isInvalid method. It should be isInvalid instead of isValid.

**Test Case 3:**

Enter the username:

123abc

**Output:**

Invalid Username

**Explanation:**

* ‘123abc’ contains non-alphabetic characters, so it is invalid.