Name : v.soumya

Batch:15

Enrollment no:2403a51390

# Assignment-8.2

# Task1:

**Promp**t: write a python program to generate test cases for a function is\_prime(n) and then implement the function takes an integer n as input and returns True if it's a prime number, and False otherwise.

**Explanation**: prime numbers are greater than 1. So, if the input number n is 1 or less (including negative numbers and 0), the function immediately returns False because these numbers are not prime. 2. This step checks if the number n is divisible by 2 using the modulo operator (%). If n is greater than 2 and divisible by 2, it means it's an even number greater than 2, and therefore it cannot be prime. In this case, the function returns False. These initial checks quickly handle several common cases and improve the efficiency of the function by avoiding unnecessary calculations for numbers that are clearly not prime.

**Code:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Task2:**

**Write a python program to generate the test cases for celsius\_to\_fahrenheit(c) and fahrenheit\_to\_celsius(f).and valid input pairs are 0°C = 32°F, 100°C = 212°F. Include decimals and invalid inputs like strings or None**

* **Explanation: If the input is valid, it applies the formula (celsius \* 9/5) + 32 to convert Celsius to Fahrenheit and returns the result.** **If the input is valid, it applies the formula (fahrenheit - 32) \* 5/9 to convert Fahrenheit to Celsius and returns the result For Celsius to Fahrenheit tests (the first 6 tuples), each tuple is (input\_celsius, expected\_fahrenheit).For Fahrenheit to Celsius tests (the last 6 tuples), each tuple is (input\_fahrenheit, expected\_celsius) For each test case, it prints the input, the expected output, and the actual output returned by the function, making it easy to see if the functions are working correctly.**

**Code:**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Task3:**

**Prompt:**

**write a python program to write test cases for a function count \_words(text) that returns the number of words in a sentence. Handle normal text, multiple spaces, punctuation, and empty strings.**

**Explanation:** **The count \_words function takes text, removes punctuation, splits the text into words based on spaces, and returns the number of words. It also checks if the input is actually a string. The test cases demonstrate how the function handles different inputs like normal text, extra spaces, punctuation, and invalid inputs.**

**Code:**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**A computer code with text

AI-generated content may be incorrect.**

**Output:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Task4:**

**Prompt: write a python program to generate a test \_cases for a Bank Account class deposit(amount), withdraw(amount), check \_balance().to give the code based on these requirments . Negative deposits/withdrawals should raise an error and Cannot withdraw more than balance.**

**Explanation:**

**The code defines a BankAccount class with methods for depositing, withdrawing, and checking the balance. It includes checks to prevent negative initial balances, negative deposits/withdrawals, and withdrawals exceeding the current balance. The test cases demonstrate these functionalities and error handling.**

**Code:**

**A computer screen shot of a code

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a computer code

AI-generated content may be incorrect.**

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Task5:**

**Prompt:** **write a python program to generate a test cases if is\_ number \_palindorme (num), which checks if an integer reads the same backward. for example the number is 121 condition true or 123 it get false otherwise 0, negative numbers handled gracefully.**

**Explanation:**

**The is \_number \_palindrome function checks if an integer is a palindrome by converting it to a string and comparing it to its reverse. It returns True for palindromes (like 121 or 0), False for non-palindromes and negative numbers, and an error message for non-integer inputs. The test cases verify these scenarios.**

**Code:**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**