|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **ProgramName:**B. Tech | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **CourseCoordinatorName** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s)Name** | | | | |  | | --- | | Dr. V. Venkataramana (Co-ordinator) | | Dr. T. Sampath Kumar | | Dr. Pramoda Patro | | Dr. Brij Kishor Tiwari | | Dr.J.Ravichander | | Dr. Mohammand Ali Shaik | | Dr. Anirodh Kumar | | Mr. S.Naresh Kumar | | Dr. RAJESH VELPULA | | Mr. Kundhan Kumar | | Ms. Ch.Rajitha | | Mr. M Prakash | | Mr. B.Raju | | Intern 1 (Dharma teja) | | Intern 2 (Sai Prasad) | | Intern 3 (Sowmya) | | NS\_2 ( Mounika) | | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week4 - Tuesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:8.2**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 8: Test-Driven Development with AI – Generating and Working with Test Cases  **Lab Objectives:**   * To introduce students to test-driven development (TDD) using AI code generation tools. * To enable the generation of test cases before writing code implementations. * To reinforce the importance of testing, validation, and error handling. * To encourage writing clean and reliable code based on AI-generated test expectations.     **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Use AI tools to write test cases for Python functions and classes. * Implement functions based on test cases in a test-first development style. * Use unittest or pytest to validate code correctness. * Analyze the completeness and coverage of AI-generated tests. * Compare AI-generated and manually written test cases for quality and logic   **Task Description#1**  Use AI to generate test cases for a function is\_prime(n) and then implement the function.  **Requirements:**   * Only integers > 1 can be prime.   Check edge cases: 0, 1, 2, negative numbers, and large primes.  **Expected Output#1**   * A working prime checker that passes AI-generated tests using edge coverage.   **Task Description#2 (Loops)**   * Ask AI to generate test cases for celsius\_to\_fahrenheit(c) and fahrenheit\_to\_celsius(f).   **Requirements**   * Validate known pairs: 0°C = 32°F, 100°C = 212°F. * Include decimals and invalid inputs like strings or None   **Expected Output#2**  Dual conversion functions with complete test coverage and safe type handling  **Task Description#3**  Use AI to write test cases for a function count\_words(text) that returns the number of words in a sentence.  **Requirement**  Handle normal text, multiple spaces, punctuation, and empty strings.  **Expected Output#3**  Accurate word count with robust test case validation.  **Task Description#4**   * Generate test cases for a BankAccount class with:   **Methods:**  deposit(amount)  withdraw(amount)  check\_balance()  **Requirements:**   * Negative deposits/withdrawals should raise an error. * Cannot withdraw more than balance.   **Expected Output#4**   * AI-generated test suite with a robust class that handles all test cases.   **Task Description#5**  Generate test cases for is\_number\_palindrome(num), which checks if an integer reads the same backward.  **Examples:**  121 → True  123 → False  0, negative numbers → handled gracefully  **Expected Output#5**   * Number-based palindrome checker function validated against test cases.   **Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots**  **Evaluation Criteria:**   | **Criteria** | **Max Marks** | | --- | --- | | Task #1 | 0.5 | | Task #2 | 0.5 | | Task #3 | 0.5 | | Task #4 | 0.5 | | Task #5 | 0.5 | | **Total** | **2.5 Marks** | | | | | | | Week4 - Wednesday |  |

**Task Description#1**

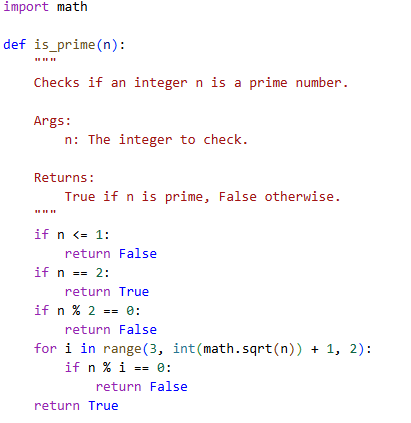
Use AI to generate test cases for a function is\_prime(n) and then implement the function.

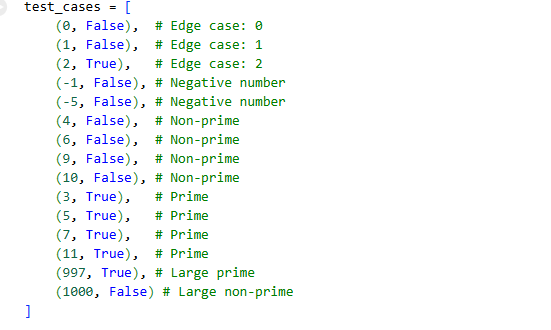
**Requirements:**

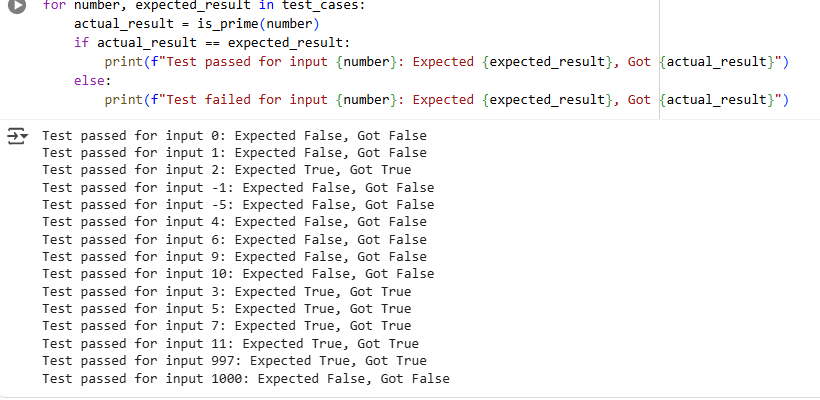
* Only integers > 1 can be prime.

Check edge cases: 0, 1, 2, negative numbers, and large primes.

**Expected Output#1**

****





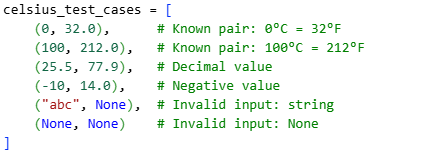
**Task Description#2 (Loops)**

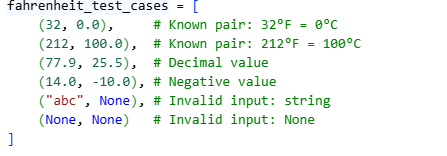
* Ask AI to generate test cases for celsius\_to\_fahrenheit(c) and fahrenheit\_to\_celsius(f).

**Requirements**

* Validate known pairs: 0°C = 32°F, 100°C = 212°F.
* Include decimals and invalid inputs like strings or None

**Expected Output#2**





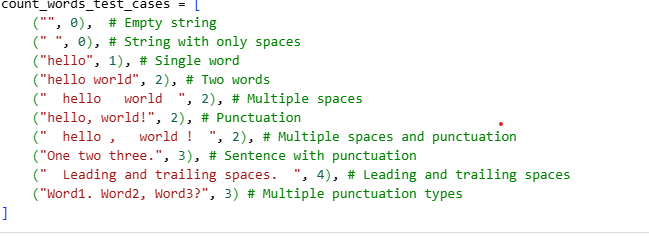
**Task Description#3**

Use AI to write test cases for a function count\_words(text) that returns the number of words in a sentence.

**Requirement**

Handle normal text, multiple spaces, punctuation, and empty strings.

**Expected Output#3**



**Task Description#4**

* Generate test cases for a BankAccount class with:

**Methods:**

deposit(amount)

withdraw(amount)

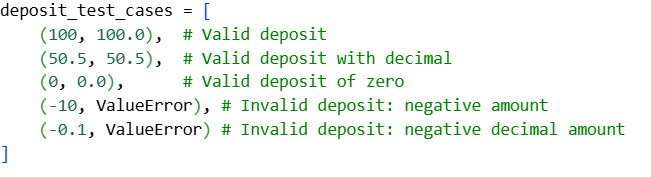
check\_balance()

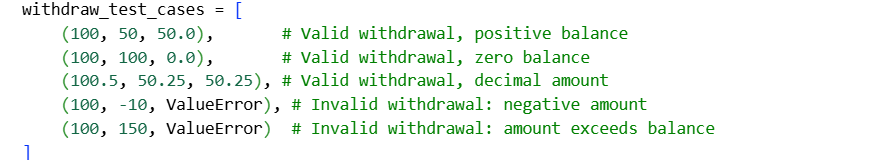
**Requirements:**

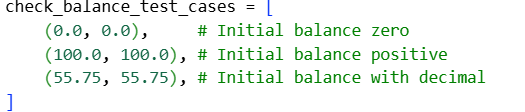
* Negative deposits/withdrawals should raise an error.
* Cannot withdraw more than balance.

**Expected Output#4**











**Task Description#5**

Generate test cases for is\_number\_palindrome(num), which checks if an integer reads the same backward.

**Examples:**

121 → True

123 → False

0, negative numbers → handled gracefully

**Expected Output#5**

