TARMEEZ
نادي ترميز

Easy

Lab 1
Number Guessing Game

Program Explanation:
This program generates a random number between 1 and 100 and asks the user to guess it. The program provides feedback if the guessed number is too high or too low. The user has a limited number of attempts, and the game can be replayed.

Test Cases:
Test the program with correct guesses within the allowed attempts.
Input: User guesses: 50, 75, 60 (random number was 60)
Expected Output: "Congratulations! You guessed the correct number in 3 attempts."

Test the program with incorrect guesses.
Input: User guesses: 80, 90, 95 (random number was 70)
Expected Output: "Too high! Try again. Attempts left: 2 / 1 / 0. The correct number was 70."

Verify that the game restarts after completion.
Input: User plays the game again, guesses: 40, 30, 35 (random number was 35)
Expected Output: "Congratulations! You guessed the correct number in 3 attempts."

Lab Hints: Remember to use the random module for generating a random number. Utilize a loop to allow the user to play the game multiple times.

TARMEEZ
نادي ترميز

Medium

Lab 2
Temperature Converter

Program Explanation:
This program converts temperatures between Celsius and Fahrenheit based on user input. The user can choose the conversion type, and the program handles non-numeric input.

Test Cases:
Test the program with a valid Celsius to Fahrenheit conversion.
Input: Celsius temperature: 25, Conversion type: 1
Expected Output: "25 degrees Celsius is equal to 77.0 degrees Fahrenheit."

Test the program with a valid Fahrenheit to Celsius conversion.
Input: Fahrenheit temperature: 98.6, Conversion type: 2
Expected Output: "98.6 degrees Fahrenheit is equal to 37.0 degrees Celsius."

Test the program with non-numeric input.
Input: Celsius temperature: "abc", Conversion type: 1
Expected Output: "Invalid input. Please enter a numeric value for the temperature."

Lab Hints: Utilize functions for code modularity. Implement error handling using try and except blocks for non-numeric input.

TARMEEZ
نادي ترميز

Hard

Lab 3
Simple Billing System

Program Explanation:
This program presents a menu to the user with the following options:
1. Read Bills: Displays all bills and allows the user to select a bill to read.
2. Write Bill: Enables the user to create a new bill by entering product details.
3. Exit: Terminates the program.

Test Cases:
Test the program by reading an existing bill.
Input: Existing bill name: "sample.txt"
Expected Output: Contents of "sample.txt" displayed on the console.

Test the program by trying to read a non-existent bill.
Input: Non-existent bill name: "nonexistent.txt"
Expected Output: "Error: Bill 'nonexistent.txt' not found. Please enter a valid bill name."

Test the program by creating a new bill, entering product details, and saving it.
Input: New bill name: "new_bill.txt", Product details: "Product: $Price"
Expected Output: "Bill 'new_bill.txt' created and saved successfully."

Lab Hints: Use the open() function for file operations.
Use the write() function to write to a file.
Use dictionaries to store product details.
Implement error handling for file not found situations.
Organize code into functions for better structure.