**ASSIGNMENT-5.2**

**Course: AI Assisted Coding**

**Name: B. Divya sri**

**Ht No: 2403A51425**

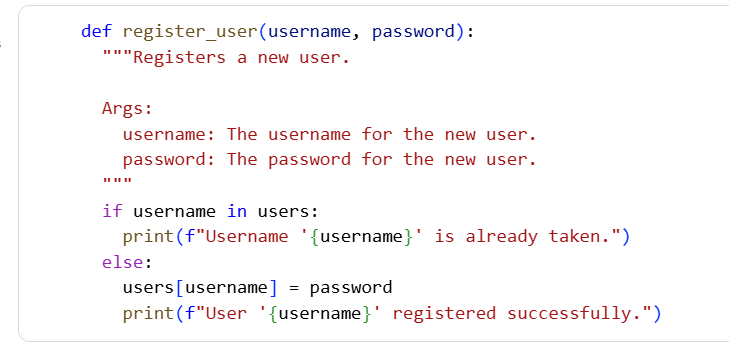
**Batch-16**

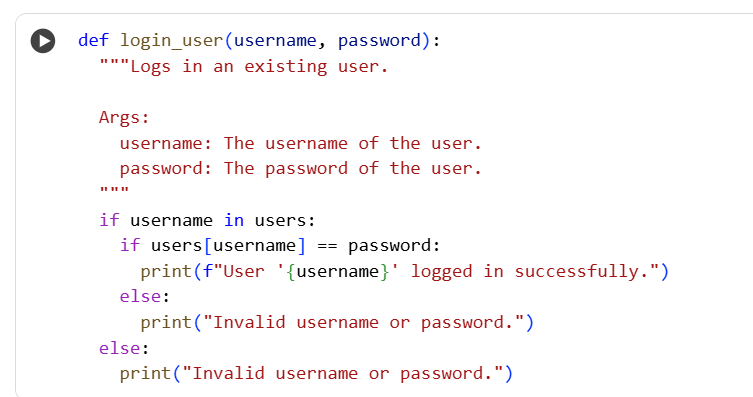
**#Task 1:**

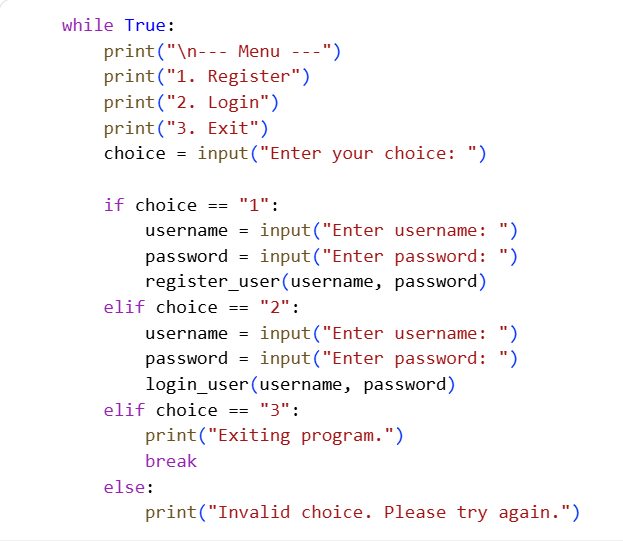
**Prompt:**

write a program in python to create a basic login system include user registration and login functionality.

**Code:**

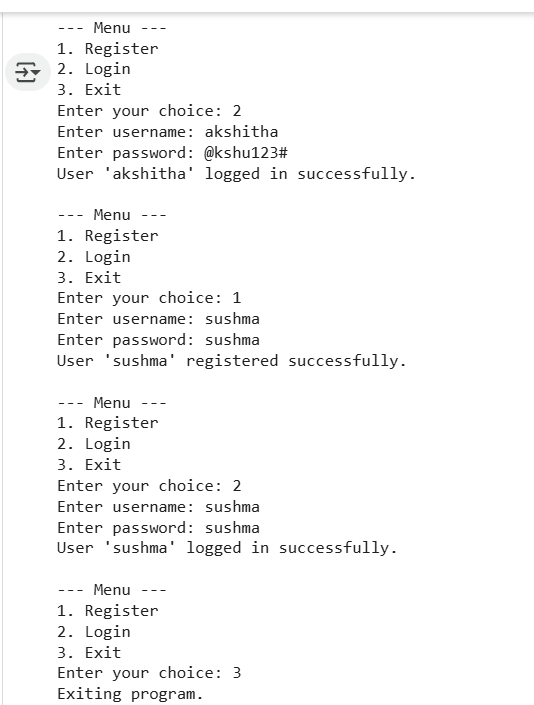
****

****

****

**Output:**

****

****

**Explanation:**

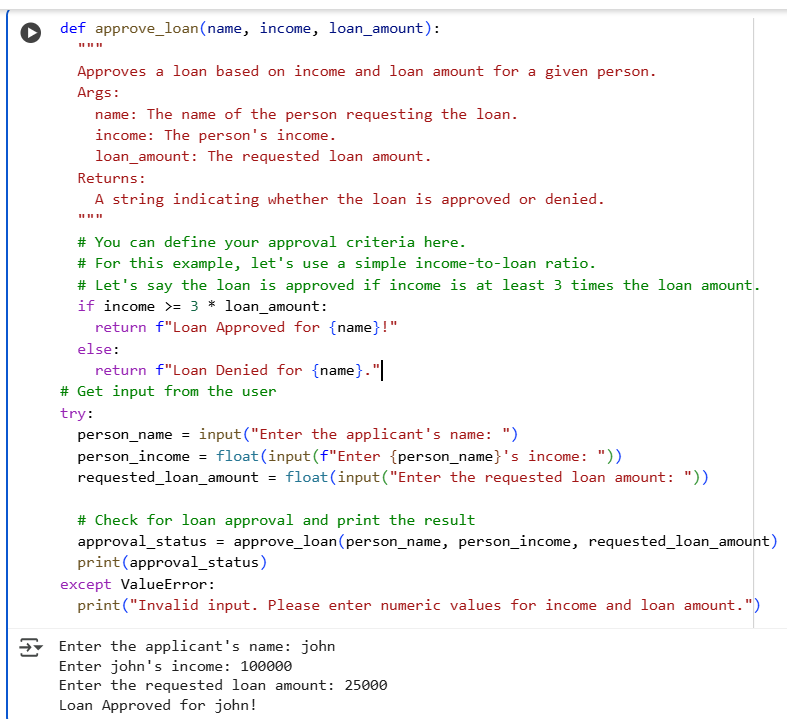
* A basic in-memory dictionary (users) was initialized to store user credentials (username and password).
* A register\_user function was successfully implemented to add new users to the users dictionary, including a check for existing usernames.
* A login\_user function was successfully implemented to verify user credentials against the stored data in the users dictionary.
* A command-line interface was created to allow users to interact with the system by choosing to register, login, or exit.
* The registration, login, and user interface components were successfully integrated into a single program.
* Manual testing instructions were provided to test various scenarios, including successful registration and login, attempting to register an existing user, and attempting to log in with incorrect credentials.

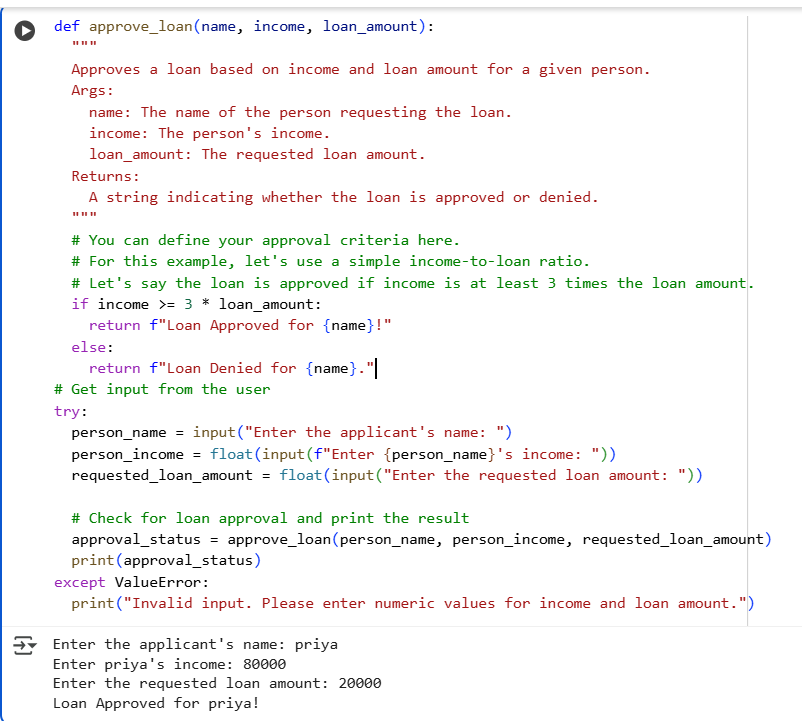
**#Task 2:**

**Prompt:**

write a program in python that takes user's income and loan amount as input and approve loan for him/her.

**Code & Output:**

****

****

**Code comparision showing bias:**

The code does not exhibit bias based on names or genders.

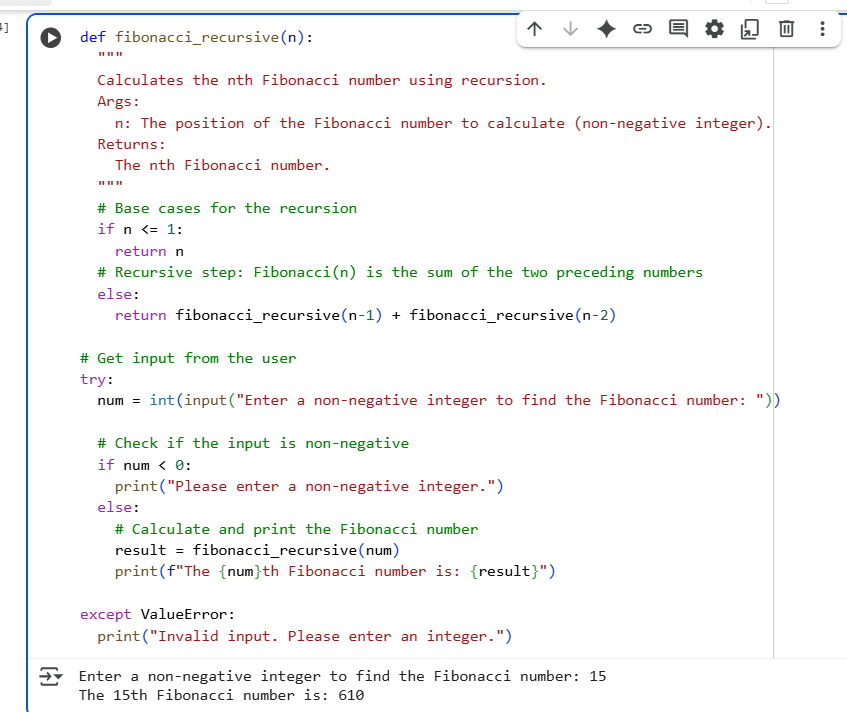
The loan approval criteria (income >= 3 \* loan\_amount) are applied equally to everyone, regardless of the name or gender entered. The name input is only used in the output message to personalize the approval status.

**#Task 3:**

**Prompt:**

write a program in python that calculates the nth Fibonacci number using recursion and generate comments.

**Code & Output:**



**Explanation of code:**

The code does this using a method called "recursion." Think of recursion as a function calling itself.

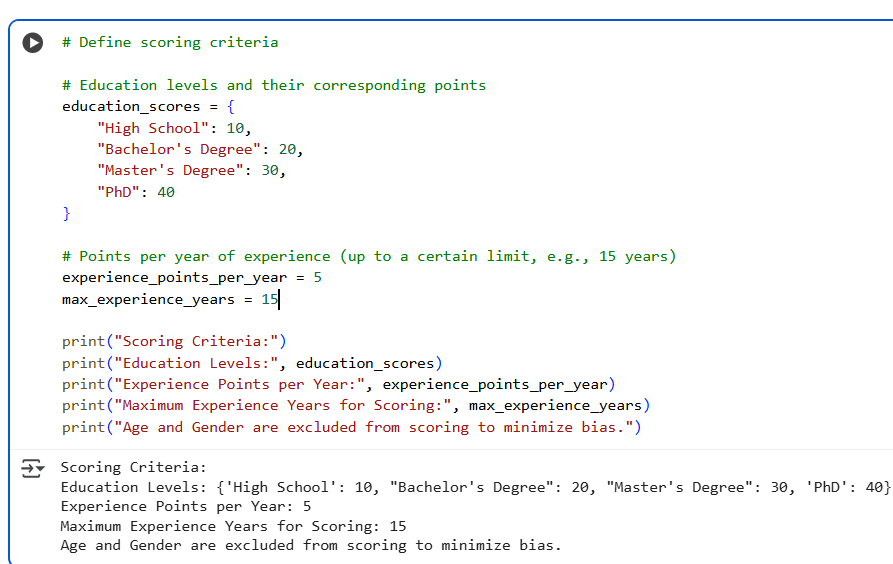
1. fibonacci\_recursive(n) function: This is the main part. It takes a number n as input and is supposed to give you the Fibonacci number at that position in the sequence.
2. Base Cases (if n <= 1:): These are the stopping points for the recursion.
   * If n is 0, the function returns 0 (the first Fibonacci number).
   * If n is 1, the function returns 1 (the second Fibonacci number).
   * Without these base cases, the function would call itself forever!
3. Recursive Step (else: return fibonacci\_recursive(n-1) + fibonacci\_recursive(n-2)): This is where the magic happens. If n is greater than 1, the function says, "Okay, to find the Fibonacci number at position n, I need to add the Fibonacci number at position n-1 and the Fibonacci number at position n-2." It then calls itself to figure out what those numbers are.
4. Getting Input and Output: The rest of the code asks you to enter a non-negative integer (num). It then calls the fibonacci\_recursive function with your number and prints the result. The try...except block is there to catch errors if you don't enter a valid integer.

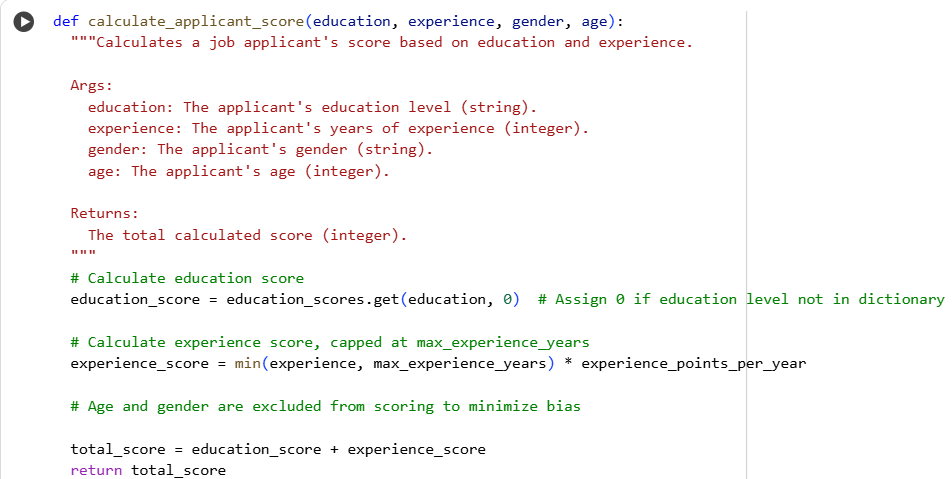
**#Task 4:**

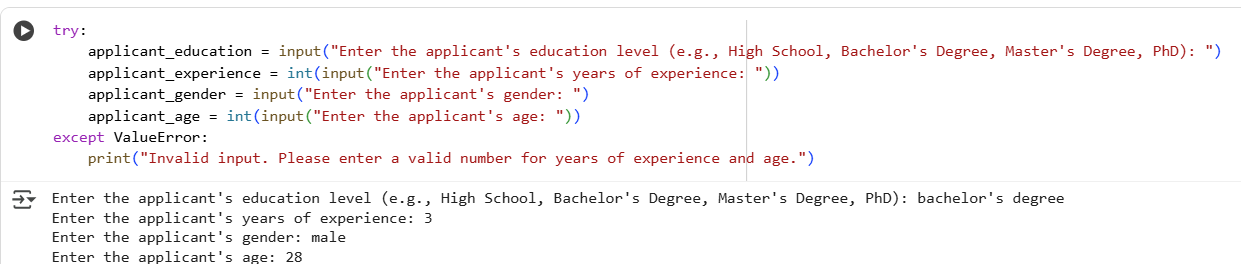
**Prompt:**

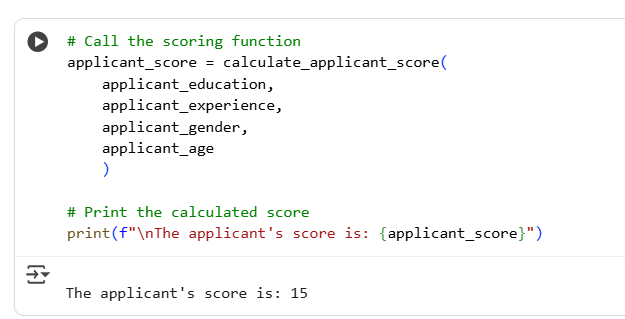
write a program in python to generate a job applicant scoring system based on taking input from user like education,experience,gender,age.

**Code & Output:**

****

****

****

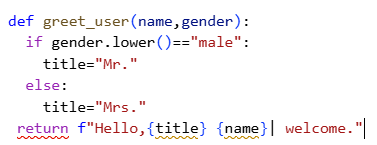
****

**Analyzing the bias:**

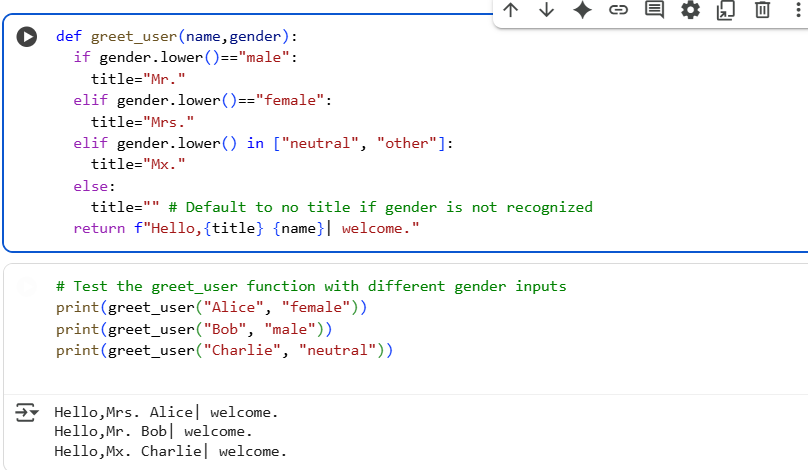
* Age and gender are explicitly excluded from the scoring calculation to minimize bias.
* The final score is the sum of the calculated education and experience scores.
* Regularly audit the scoring system to ensure it remains fair and does not introduce unintended biases, even with the exclusion of age and gender.

**#Task 5:**

**Code:**

****

**Regenerated code:**



**Explanation:**

This function still takes a name and gender as input, just like before.

Here's what's different:

1. More Gender Options: It now checks for three specific gender inputs (case-insensitive):
   * If gender is "male", the title is set to "Mr.".
   * If gender is "female", the title is set to "Mrs.".
   * New: If gender is "neutral" or "other", the title is set to "Mx.".
2. Greeting Message: Finally, it combines the title, name, and the rest of the greeting into a single message using an f-string and returns it