ASSIGNMENT 5.3

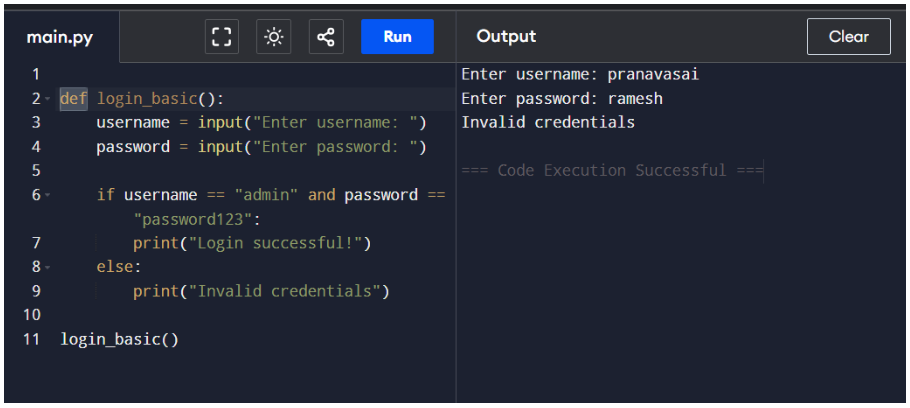
2303A51695 P. Nandhini BATCH-28

Ethical and Secure Use of AI-Generated Code

Task 1: Privacy and Data Security in AI-Generated Code

PROMPT: GENERATE A LOGIN CODE IN PYTHON

CODE AND OUTPUT:



```python

def login_basic():
    username = input("Enter username: ")
    password = input("Enter password: ")

    if username == "admin" and password == "password123":
        print("Login successful!")
    else:
        print("Invalid credentials")

login_basic()
```

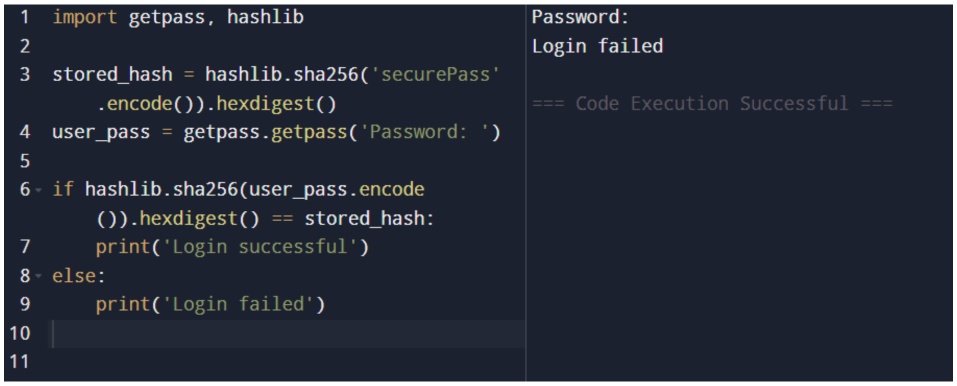
```
Enter username: pranavasai
Enter password: ramesh
Invalid credentials

=== Code Execution Successful ===
```

Security Issues Identified:
- Credentials are hardcoded
- Password is stored and compared in plain text
- No input validation or hashing

PROMPT: GENERATE A SECURE CODE

CODE AND OUTPUT:



```python
import getpass, hashlib

stored_hash = hashlib.sha256('securePass'.encode()).hexdigest()
user_pass = getpass.getpass('Password: ')

if hashlib.sha256(user_pass.encode()).hexdigest() == stored_hash:
    print('Login successful')
else:
    print('Login failed')


```

```
Password:
Login failed

=== Code Execution Successful ===
```

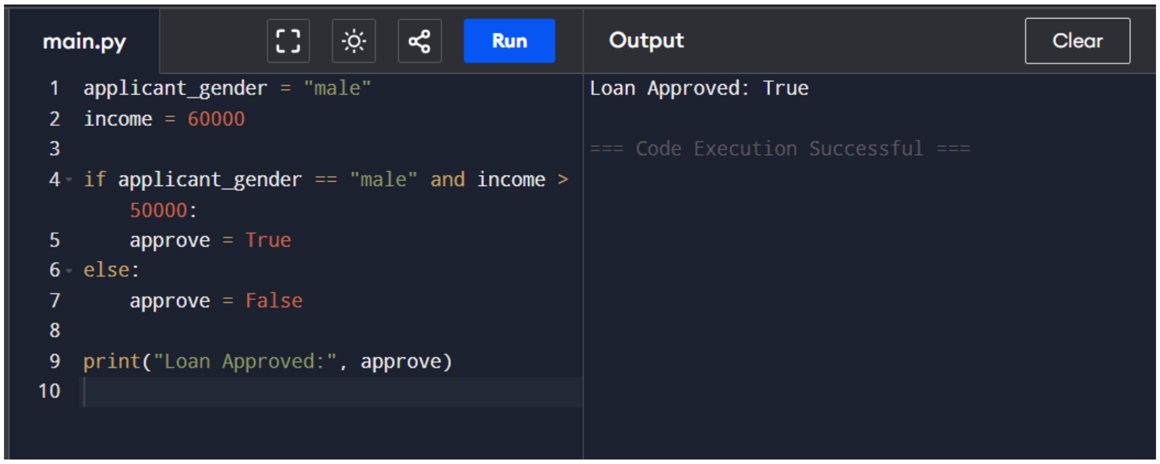
Improvements:
- Removed hardcoded credentials
- Used password hashing
- Used secure password input

Task 2: Bias Detection in AI-Generated Decision Systems

PROMPT:

Generate A Loan Approval Code (Biased Example):

CODE AND OUTPUT:



```python
applicant_gender = "male"
income = 60000

if applicant_gender == "male" and income > 50000:
    approve = True
else:
    approve = False

print("Loan Approved:", approve)
```

```
Loan Approved: True

=== Code Execution Successful ===
```

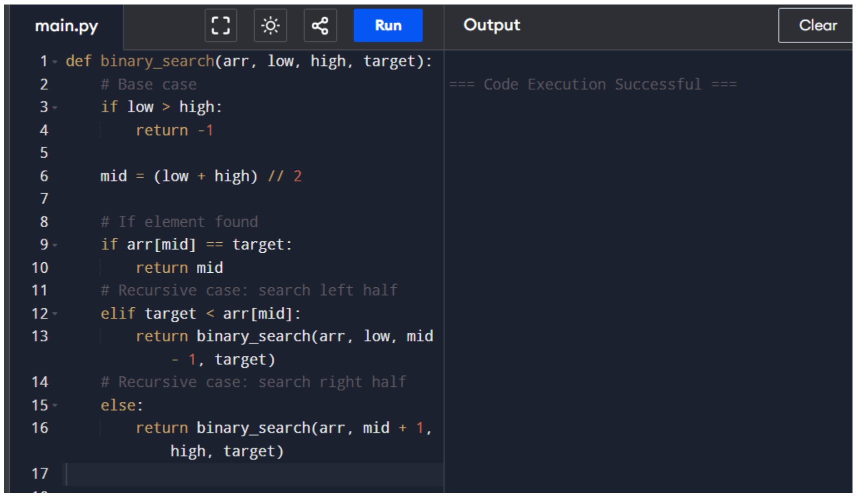
Bias Identified:
- Approval depends on gender
- Gender is irrelevant to loan eligibility

Fairness Discussion:
Such logic unfairly disadvantages applicants based on gender.

Mitigation Strategies:
- Remove gender from decision logic
- Use only financial criteria
- Regular bias audits

Task 3: Transparency and Explainability (Recursive Binary Search)

PROMPT: Generate a Recursive Binary Search Code:



```
def binary_search(arr, low, high, target):
    # Base case
    if low > high:
        return -1

    mid = (low + high) // 2

    # If element found
    if arr[mid] == target:
        return mid
    # Recursive case: search left half
    elif target < arr[mid]:
        return binary_search(arr, low, mid - 1, target)
    # Recursive case: search right half
    else:
        return binary_search(arr, mid + 1, high, target)
```

```
=== Code Execution Successful ===
```

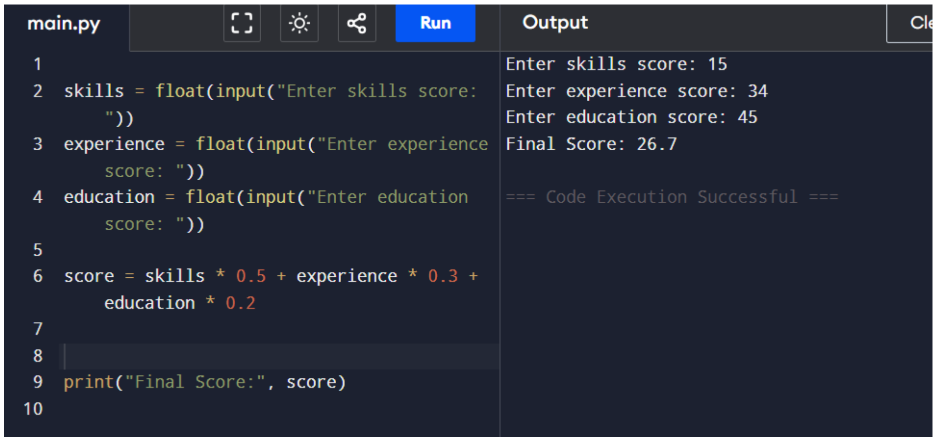
Code and Output:

Explanation:
- Base case stops recursion when search range is invalid
- Recursive case reduces problem size each call
Code is beginner-friendly, well-commented, and logically clear.

Task 4: Ethical Evaluation of AI-Based Scoring Systems

PROMPT: Generate a Scoring Code:

CODE AND OUTPUT:



```

skills = float(input("Enter skills score: "))
experience = float(input("Enter experience score: "))
education = float(input("Enter education score: "))

score = skills * 0.5 + experience * 0.3 + education * 0.2


print("Final Score:", score)

```

```
Enter skills score: 15
Enter experience score: 34
Enter education score: 45
Final Score: 26.7

=== Code Execution Successful ===
```

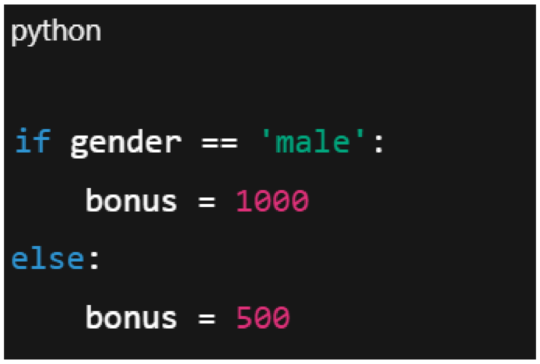
Bias Analysis:
- No gender or name used
- Logic is objective and skill-based

Ethical Evaluation:
Scoring is fair as it relies on job-relevant attributes.

Task 5: Inclusiveness and Ethical Variable Design

PROMPT: generate a code for bonus based on if gender is male

CODE:



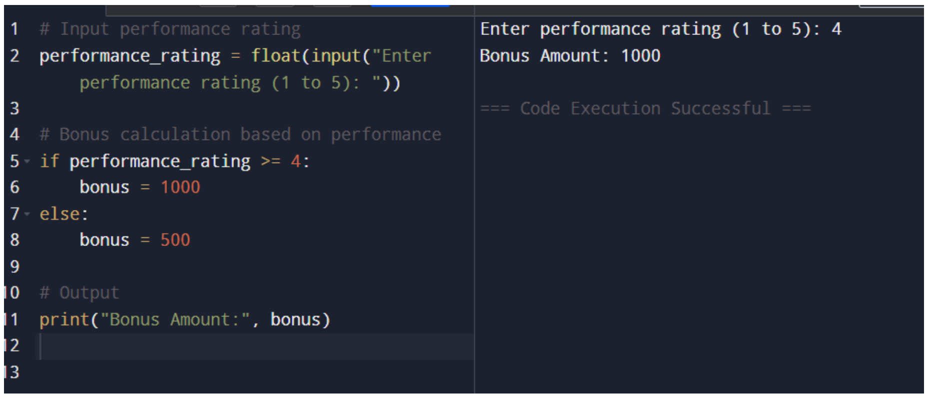
```python
if gender == 'male':
    bonus = 1000
else:
    bonus = 500
```

Issues:
- Gender-based assumptions
- Non-inclusive logic

Revised Inclusive Code:

PROMPT: GENERATE A CODE BASED ON PERFORMANCE

CODE AND OUPUT:



```python
# Input performance rating
performance_rating = float(input("Enter performance rating (1 to 5): "))

# Bonus calculation based on performance
if performance_rating >= 4:
    bonus = 1000
else:
    bonus = 500

# Output
print("Bonus Amount:", bonus)
```

```
Enter performance rating (1 to 5): 4
Bonus Amount: 1000

=== Code Execution Successful ===
```

Improvements:
- Removed gender dependency
- Used performance-based logic