ASSIGNMENT-6.5

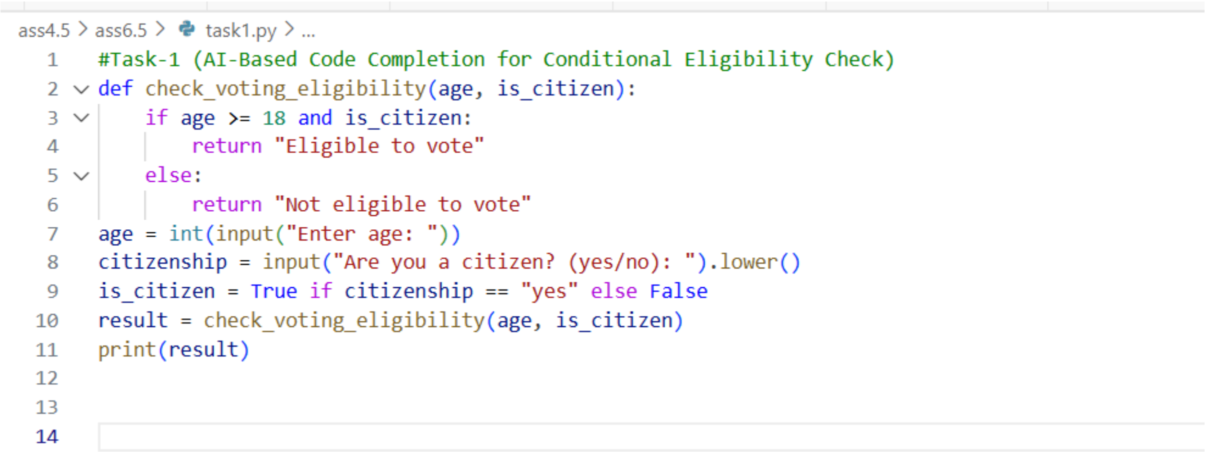
HT.NO:2303A510I4

Batch.No:30

Task 1:

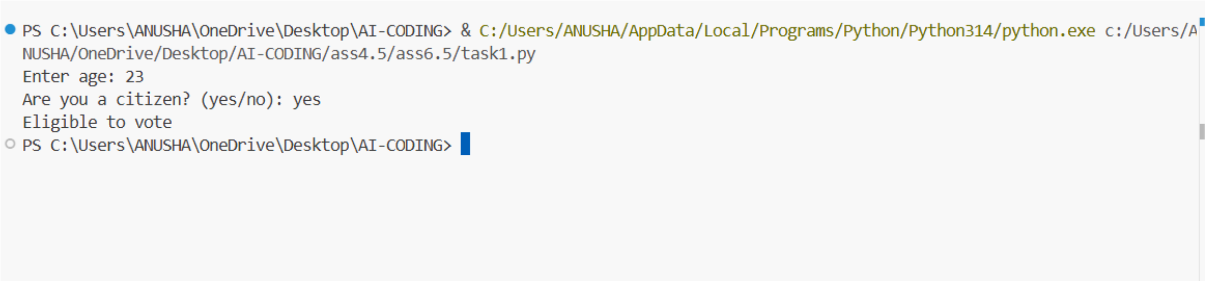
Use an AI tool to generate eligibility logic. Prompt: Generate a Python script that validates voter eligibility based on minimum age and citizenship status.

Code:



```python
#Task-1 (AI-Based Code Completion for Conditional Eligibility Check)
def check_voting_eligibility(age, is_citizen):
    if age >= 18 and is_citizen:
        return "Eligible to vote"
    else:
        return "Not eligible to vote"
age = int(input("Enter age: "))
citizenship = input("Are you a citizen? (yes/no): ").lower()
is_citizen = True if citizenship == "yes" else False
result = check_voting_eligibility(age, is_citizen)
print(result)
```

OUTPUT:



```
PS C:\Users\ANUSHA\OneDrive\Desktop\AI-CODING> & C:/Users/ANUSHA/AppData/Local/Programs/Python/Python314/python.exe c:/Users/A
NUSHA/OneDrive/Desktop/AI-CODING/ass4.5/ass6.5/task1.py
Enter age: 23
Are you a citizen? (yes/no): yes
Eligible to vote
PS C:\Users\ANUSHA\OneDrive\Desktop\AI-CODING>
```

Observation:

The conditional logic correctly determines voting eligibility based on age and citizenship.

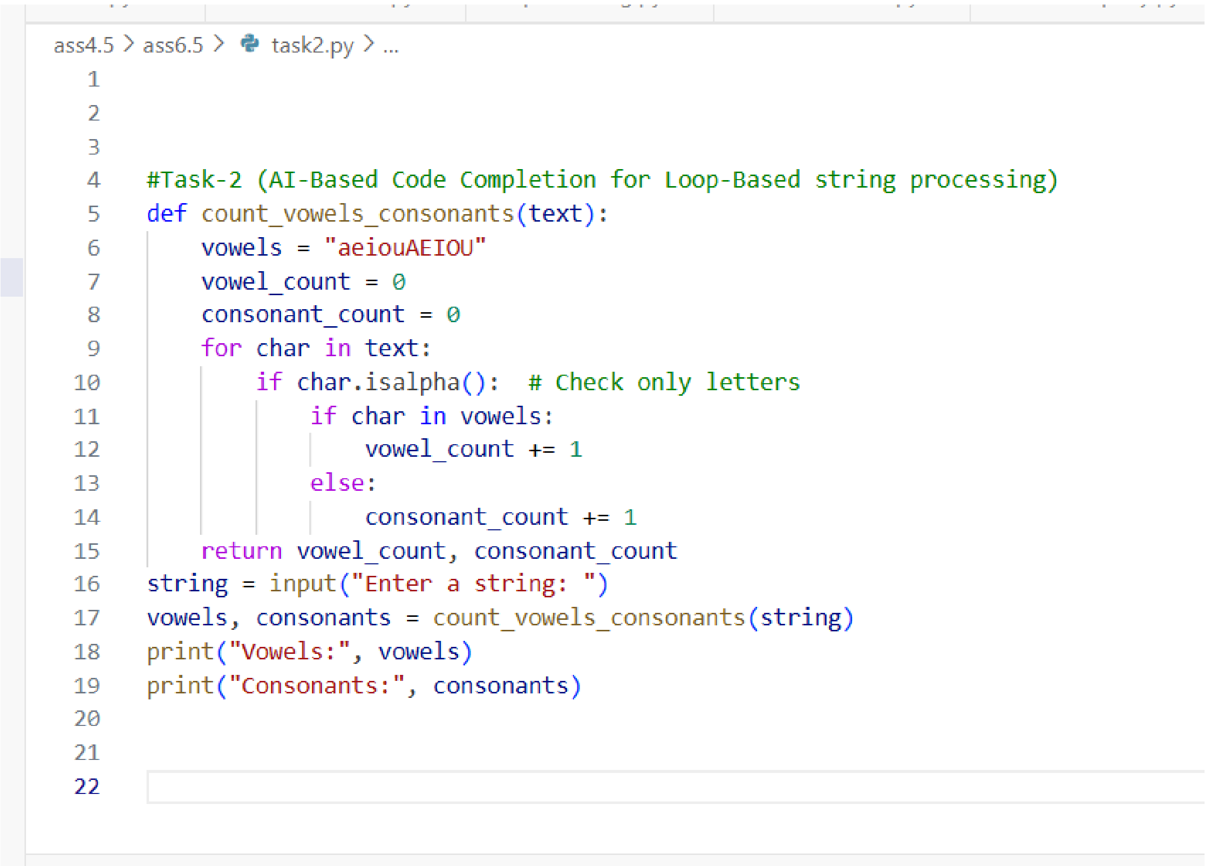
Task 2:

Use an AI tool to process strings using loops.

Prompt:

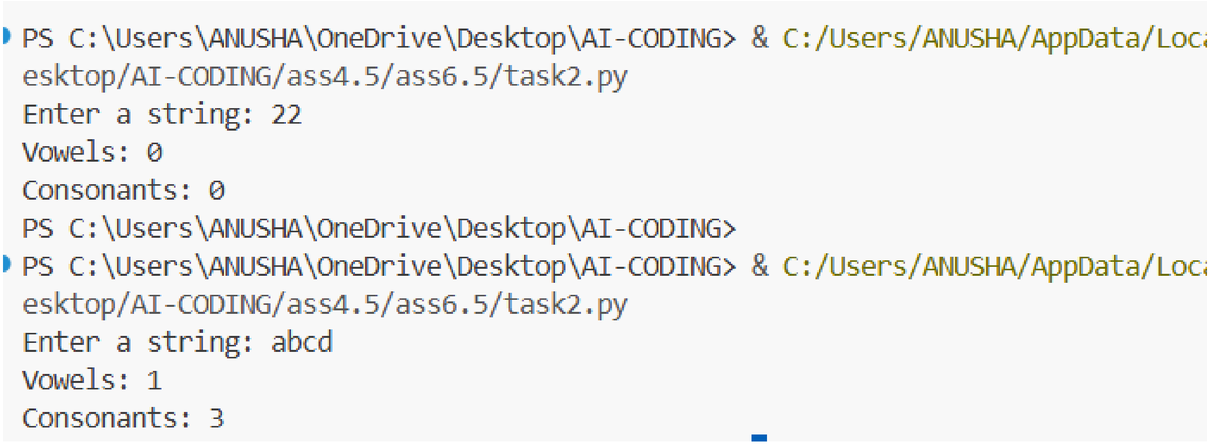
Generate a Python program that iterates through a given string and displays the total number of vowels and consonants.

Code:



```python
#Task-2 (AI-Based Code Completion for Loop-Based string processing)
def count_vowels_consonants(text):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0
    for char in text:
        if char.isalpha():  # Check only letters
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1
    return vowel_count, consonant_count
string = input("Enter a string: ")
vowels, consonants = count_vowels_consonants(string)
print("Vowels:", vowels)
print("Consonants:", consonants)
```

OUTPUT:



```
PS C:\Users\ANUSHA\OneDrive\Desktop\AI-CODING> & C:/Users/ANUSHA/AppData/Loc
esktop/AI-CODING/ass4.5/ass6.5/task2.py
Enter a string: 22
Vowels: 0
Consonants: 0
PS C:\Users\ANUSHA\OneDrive\Desktop\AI-CODING>
PS C:\Users\ANUSHA\OneDrive\Desktop\AI-CODING> & C:/Users/ANUSHA/AppData/Loc
esktop/AI-CODING/ass4.5/ass6.5/task2.py
Enter a string: abcd
Vowels: 1
Consonants: 3
```

Observation:

The program accurately processes the string and provides correct vowel and consonant counts.

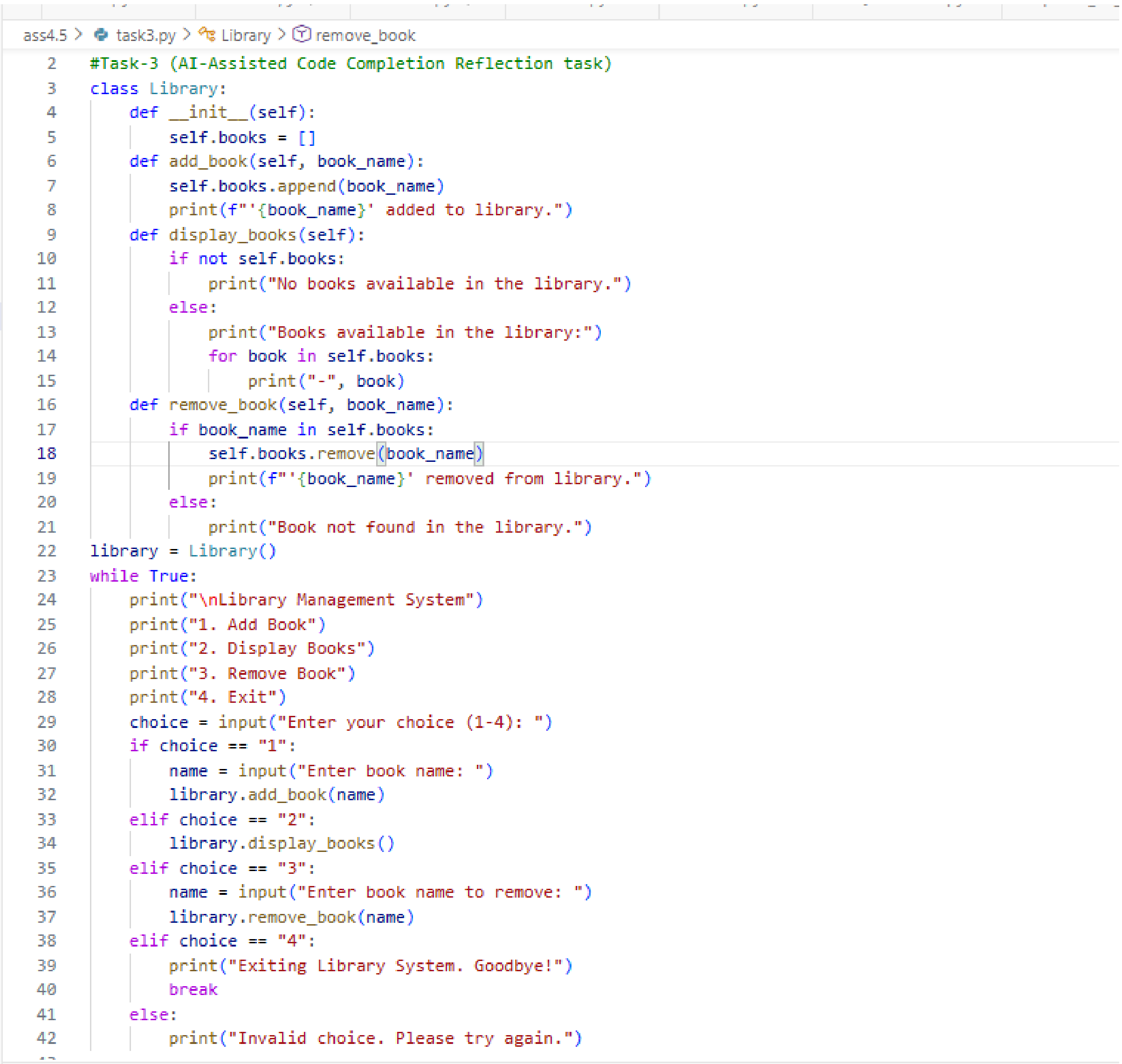
Task 3:

Use an AI tool to generate a complete program using classes, loops, and conditionals.

Prompt:

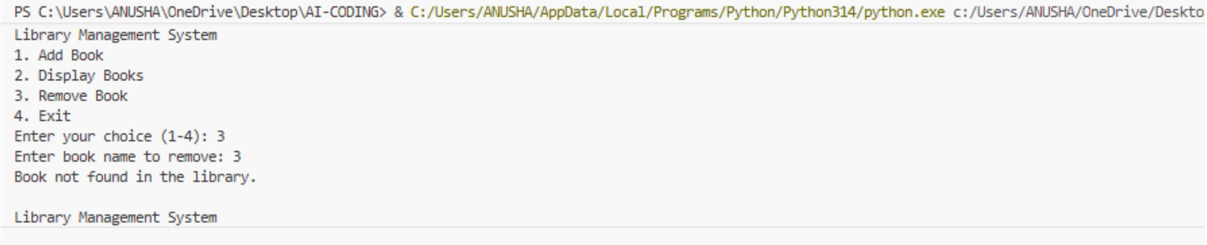
Create a complete Python program for a library management system using classes, loops, and conditional statements.

Code:



```python
#Task-3 (AI-Assisted Code Completion Reflection task)
class Library:
    def __init__(self):
        self.books = []
    def add_book(self, book_name):
        self.books.append(book_name)
        print(f"'{book_name}' added to library.")
    def display_books(self):
        if not self.books:
            print("No books available in the library.")
        else:
            print("Books available in the library:")
            for book in self.books:
                print("-", book)
    def remove_book(self, book_name):
        if book_name in self.books:
            self.books.remove(book_name)
            print(f"'{book_name}' removed from library.")
        else:
            print("Book not found in the library.")
library = Library()
while True:
    print("\nLibrary Management System")
    print("1. Add Book")
    print("2. Display Books")
    print("3. Remove Book")
    print("4. Exit")
    choice = input("Enter your choice (1-4): ")
    if choice == "1":
        name = input("Enter book name: ")
        library.add_book(name)
    elif choice == "2":
        library.display_books()
    elif choice == "3":
        name = input("Enter book name to remove: ")
        library.remove_book(name)
    elif choice == "4":
        print("Exiting Library System. Goodbye!")
        break
    else:
        print("Invalid choice. Please try again.")
```

OUTPUT:



```
PS C:\Users\ANUSHA\OneDrive\Desktop\AI-CODING> & C:/Users/ANUSHA/AppData/Local/Programs/Python/Python314/python.exe c:/Users/ANUSHA/OneDrive/Deskto
Library Management System
1. Add Book
2. Display Books
3. Remove Book
4. Exit
Enter your choice (1-4): 3
Enter book name to remove: 3
Book not found in the library.

Library Management System
```

Observation:

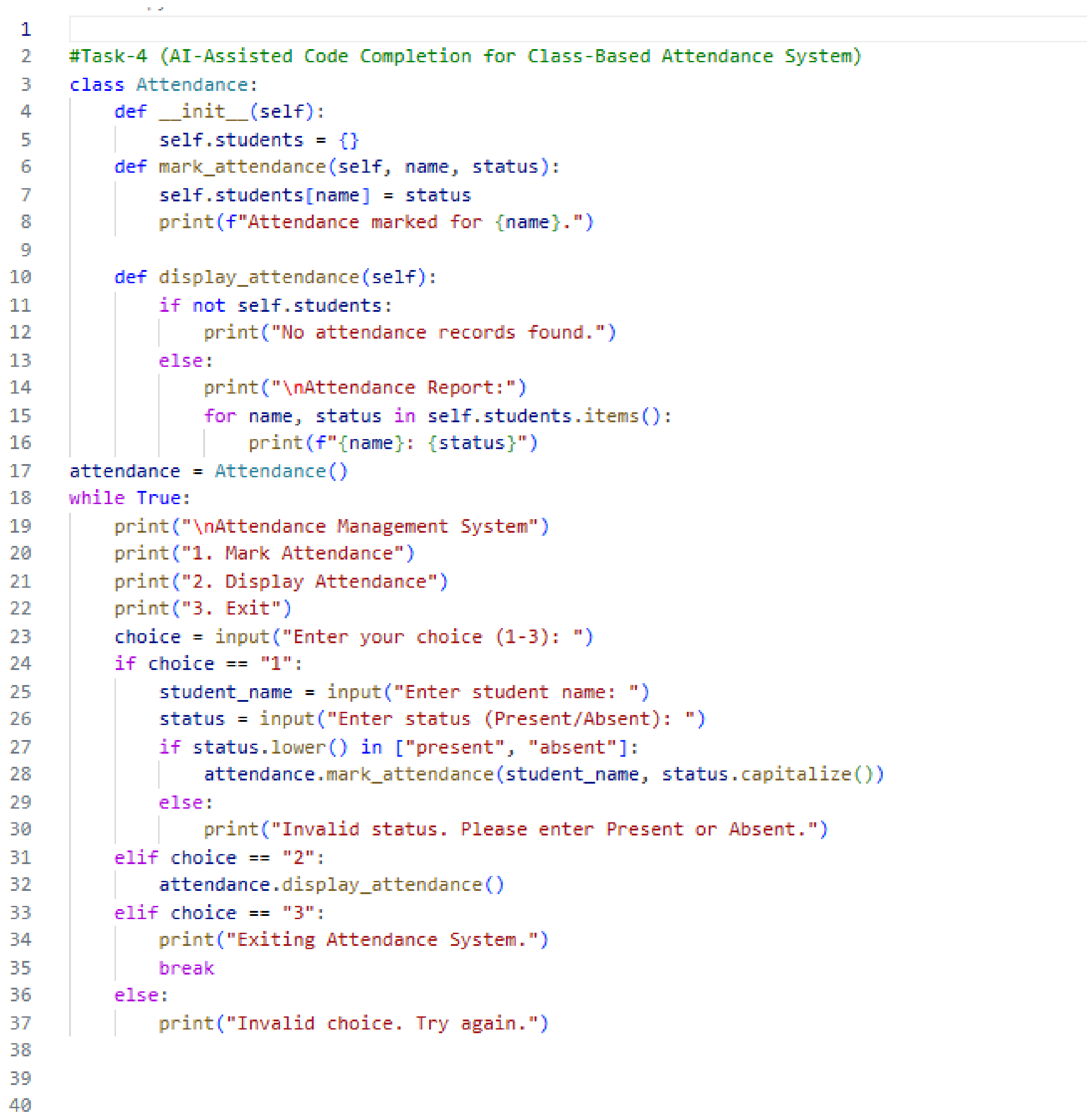
The program uses a class to organize library operations such as adding, displaying, and removing books. A menu-driven loop with conditional statements allows continuous interaction with the user. Overall, the code clearly demonstrates the practical use of classes, loops, and conditionals.

Task 4:

Use an AI tool to generate an attendance management class

Prompt: Generate a Python class that records and displays student attendance using loops.

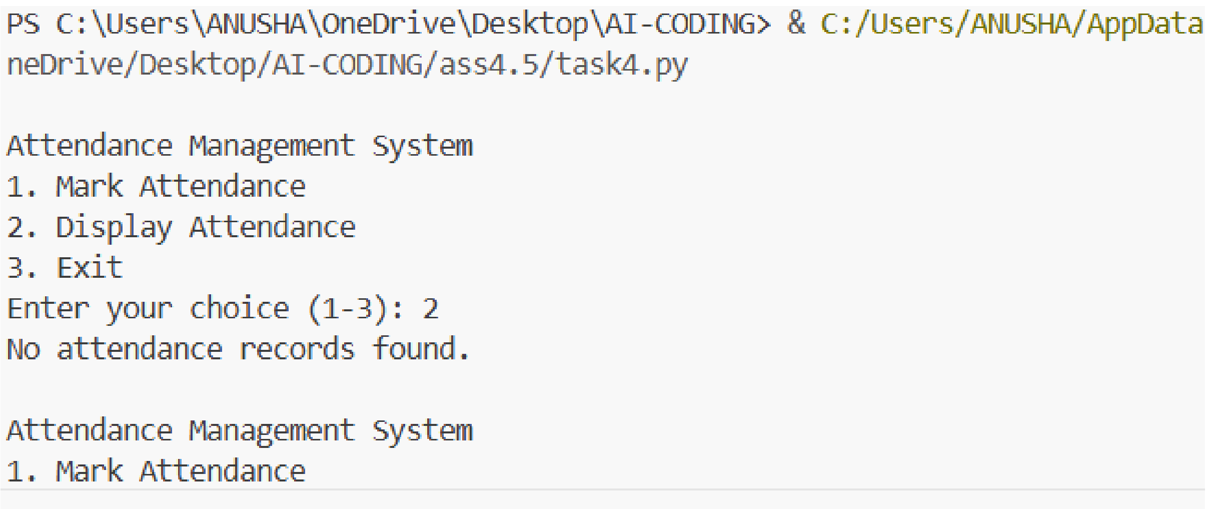
Code:



```
#Task-4 (AI-Assisted Code Completion for Class-Based Attendance System)
class Attendance:
    def __init__(self):
        self.students = {}
    def mark_attendance(self, name, status):
        self.students[name] = status
        print(f"Attendance marked for {name}.")

    def display_attendance(self):
        if not self.students:
            print("No attendance records found.")
        else:
            print("\nAttendance Report:")
            for name, status in self.students.items():
                print(f"{name}: {status}")
attendance = Attendance()
while True:
    print("\nAttendance Management System")
    print("1. Mark Attendance")
    print("2. Display Attendance")
    print("3. Exit")
    choice = input("Enter your choice (1-3): ")
    if choice == "1":
        student_name = input("Enter student name: ")
        status = input("Enter status (Present/Absent): ")
        if status.lower() in ["present", "absent"]:
            attendance.mark_attendance(student_name, status.capitalize())
        else:
            print("Invalid status. Please enter Present or Absent.")
    elif choice == "2":
        attendance.display_attendance()
    elif choice == "3":
        print("Exiting Attendance System.")
        break
    else:
        print("Invalid choice. Try again.")
```

OUTPUT:



```
PS C:\Users\ANUSHA\OneDrive\Desktop\AI-CODING> & C:/Users/ANUSHA/AppData
neDrive/Desktop/AI-CODING/ass4.5/task4.py

Attendance Management System
1. Mark Attendance
2. Display Attendance
3. Exit
Enter your choice (1-3): 2
No attendance records found.

Attendance Management System
1. Mark Attendance
```

Observation:

The program uses a class to store and manage student attendance records efficiently. A menu-driven loop with conditional statements allows marking and displaying attendance. The attendance details are displayed correctly using dictionary traversal.

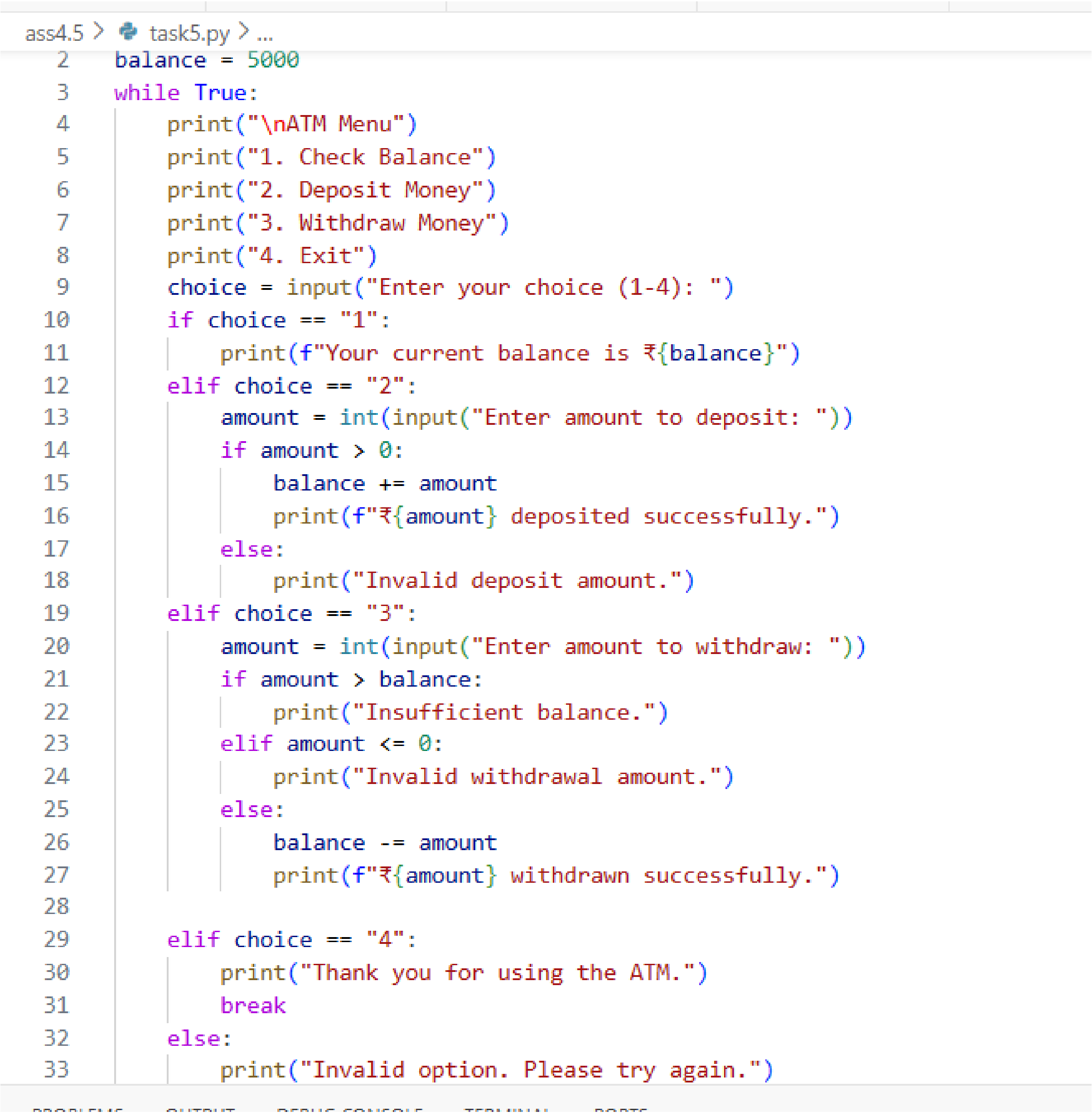
Task 5:

Use an AI tool to complete a navigation menu.

Prompt:

Develop a Python program that simulates an ATM system using loops and conditional statements.

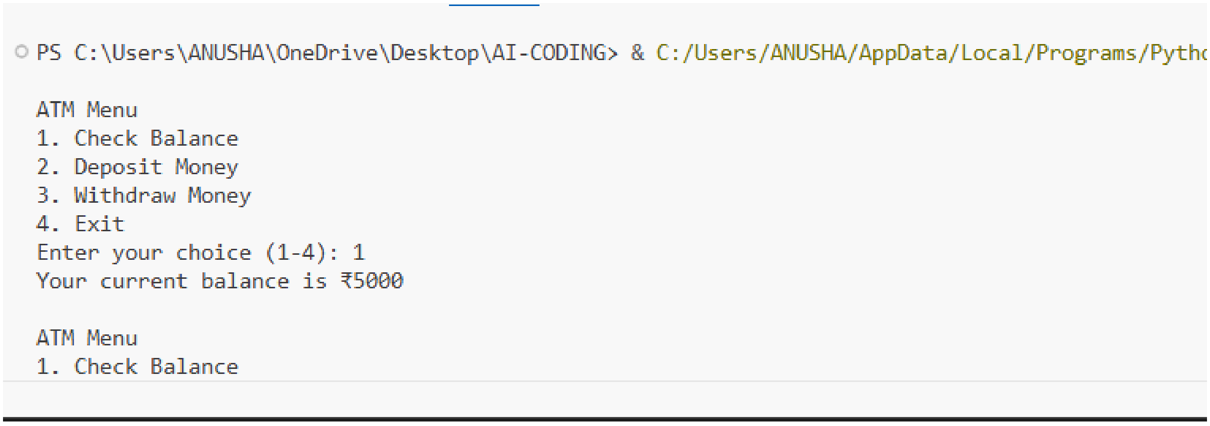
Code:



```python
balance = 5000
while True:
    print("\nATM Menu")
    print("1. Check Balance")
    print("2. Deposit Money")
    print("3. Withdraw Money")
    print("4. Exit")
    choice = input("Enter your choice (1-4): ")
    if choice == "1":
        print(f"Your current balance is ₹{balance}")
    elif choice == "2":
        amount = int(input("Enter amount to deposit: "))
        if amount > 0:
            balance += amount
            print(f"₹{amount} deposited successfully.")
        else:
            print("Invalid deposit amount.")
    elif choice == "3":
        amount = int(input("Enter amount to withdraw: "))
        if amount > balance:
            print("Insufficient balance.")
        elif amount <= 0:
            print("Invalid withdrawal amount.")
        else:
            balance -= amount
            print(f"₹{amount} withdrawn successfully.")

    elif choice == "4":
        print("Thank you for using the ATM.")
        break
    else:
        print("Invalid option. Please try again.")
```

OUTPUT:



```
○ PS C:\Users\ANUSHA\OneDrive\Desktop\AI-CODING> & C:/Users/ANUSHA/AppData/Local/Programs/Pytho

  ATM Menu
  1. Check Balance
  2. Deposit Money
  3. Withdraw Money
  4. Exit
  Enter your choice (1-4): 1
  Your current balance is ₹5000

  ATM Menu
  1. Check Balance
```

Observation:

The program uses a class to manage ATM operations such as deposit, withdrawal, and balance enquiry. A menu-driven loop with conditional statements allows continuous user interaction. Input validation is handled properly to prevent invalid transactions and insufficient balance errors.