RAVITEJA VADLURI

2303A51942

BATCH 12

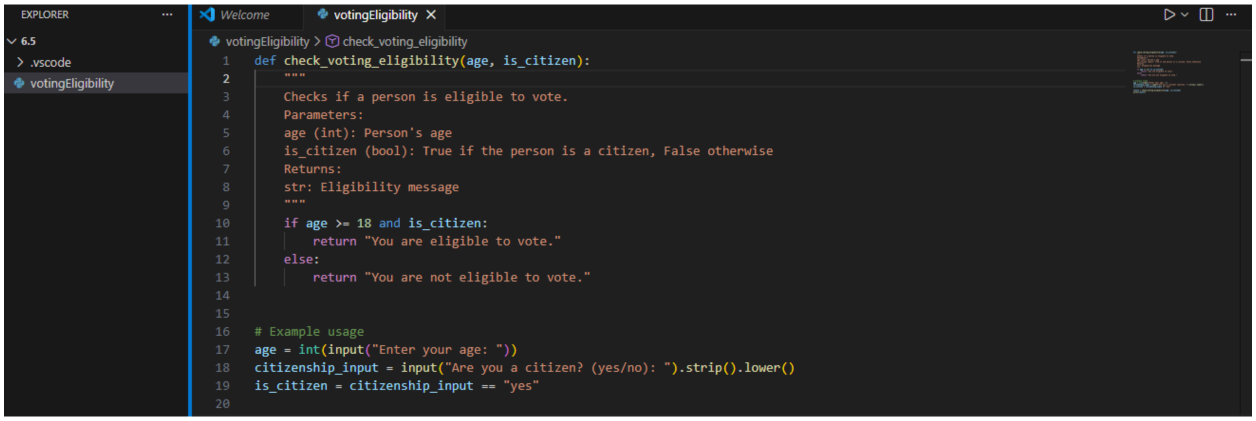
Prompt:

“Generate Python code to check voting eligibility based on age and citizenship.”

Explanation :
This program checks voting eligibility using conditional statements.
A person must be at least 18 years old to vote.
They must also be a citizen to qualify.
Both conditions must be true for eligibility.

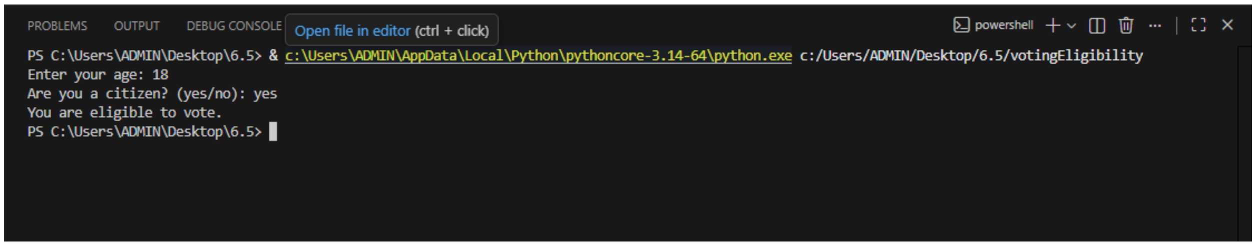
Algorithm:

```
Start
Read age and citizenship
If age ≥ 18 and citizen = yes, display eligible
Else display not eligible
Stop
```



```python
def check_voting_eligibility(age, is_citizen):
    """
    Checks if a person is eligible to vote.
    Parameters:
    age (int): Person's age
    is_citizen (bool): True if the person is a citizen, False otherwise
    Returns:
    str: Eligibility message
    """
    if age >= 18 and is_citizen:
        return "You are eligible to vote."
    else:
        return "You are not eligible to vote."


# Example usage
age = int(input("Enter your age: "))
citizenship_input = input("Are you a citizen? (yes/no): ").strip().lower()
is_citizen = citizenship_input == "yes"

```

OUTPUT



```
PS C:\Users\ADMIN\Desktop\6.5> & c:\Users\ADMIN\AppData\Local\Python\pythoncore-3.14-64\python.exe c:/Users/ADMIN/Desktop/6.5/votingEligibility
Enter your age: 18
Are you a citizen? (yes/no): yes
You are eligible to vote.
PS C:\Users\ADMIN\Desktop\6.5>
```

Task 2: Count Vowels and Consonants

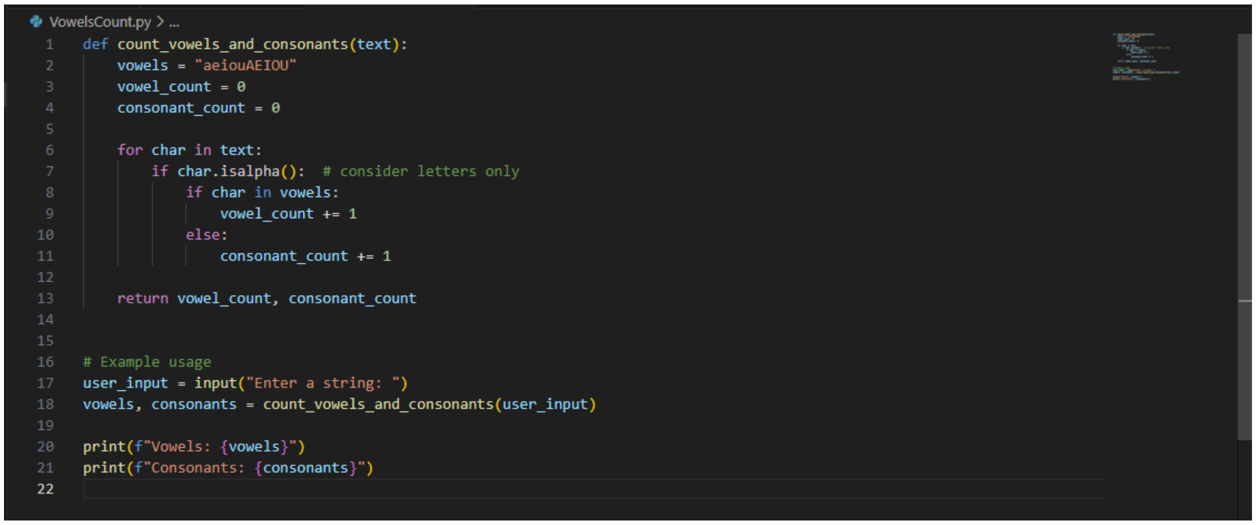
Prompt

“Generate Python code to count vowels and consonants in a string using a loop.”

Explanation :
This program counts vowels and consonants in a given string.
A loop is used to read each character one by one.
Vowels and consonants are counted separately.
Non-alphabet characters are ignored.

Algorithm:

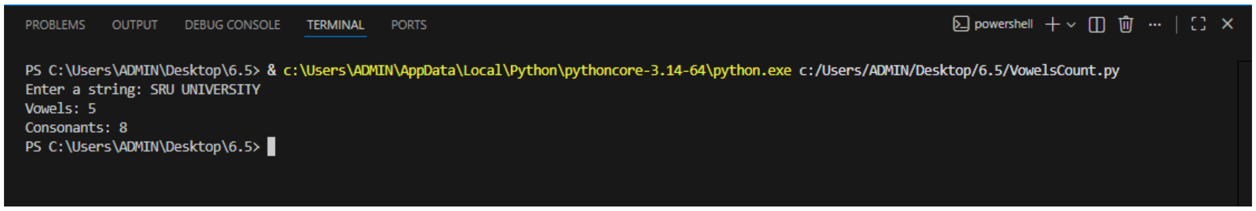
```
Start
Read a string
For each character in the string, check vowel or consonant
Display vowel and consonant count
Stop
```



VowelsCount.py > ...

```python
def count_vowels_and_consonants(text):
    vowels = "aeiouAEIOU"
    vowel_count = 0
    consonant_count = 0

    for char in text:
        if char.isalpha():  # consider letters only
            if char in vowels:
                vowel_count += 1
            else:
                consonant_count += 1

    return vowel_count, consonant_count


# Example usage
user_input = input("Enter a string: ")
vowels, consonants = count_vowels_and_consonants(user_input)

print(f"Vowels: {vowels}")
print(f"Consonants: {consonants}")
```

OUTPUT



```
PS C:\Users\ADMIN\Desktop\6.5> & c:\Users\ADMIN\AppData\Local\Python\pythoncore-3.14-64\python.exe c:/Users/ADMIN/Desktop/6.5/VowelsCount.py
Enter a string: SRU UNIVERSITY
Vowels: 5
Consonants: 8
PS C:\Users\ADMIN\Desktop\6.5>
```

Task 3: Library Management System

Prompt:

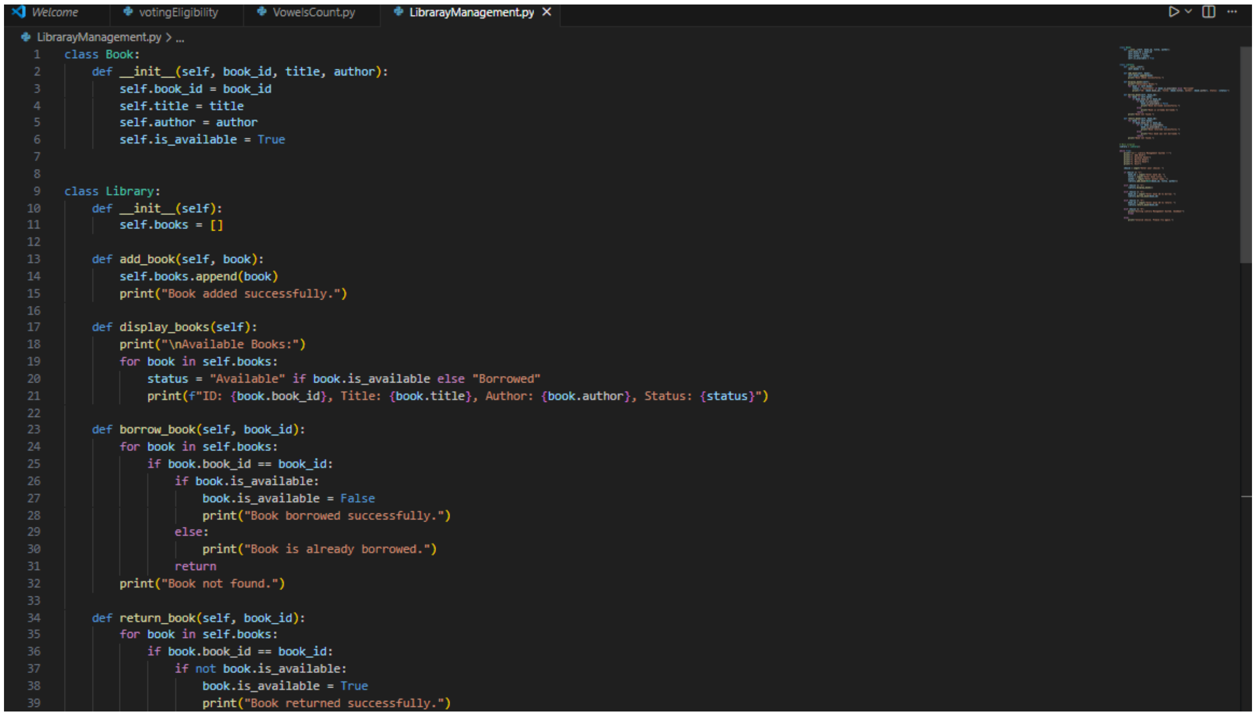
“Generate a Python program for a library management system using classes, loops, and conditional statements.”

Explanation :

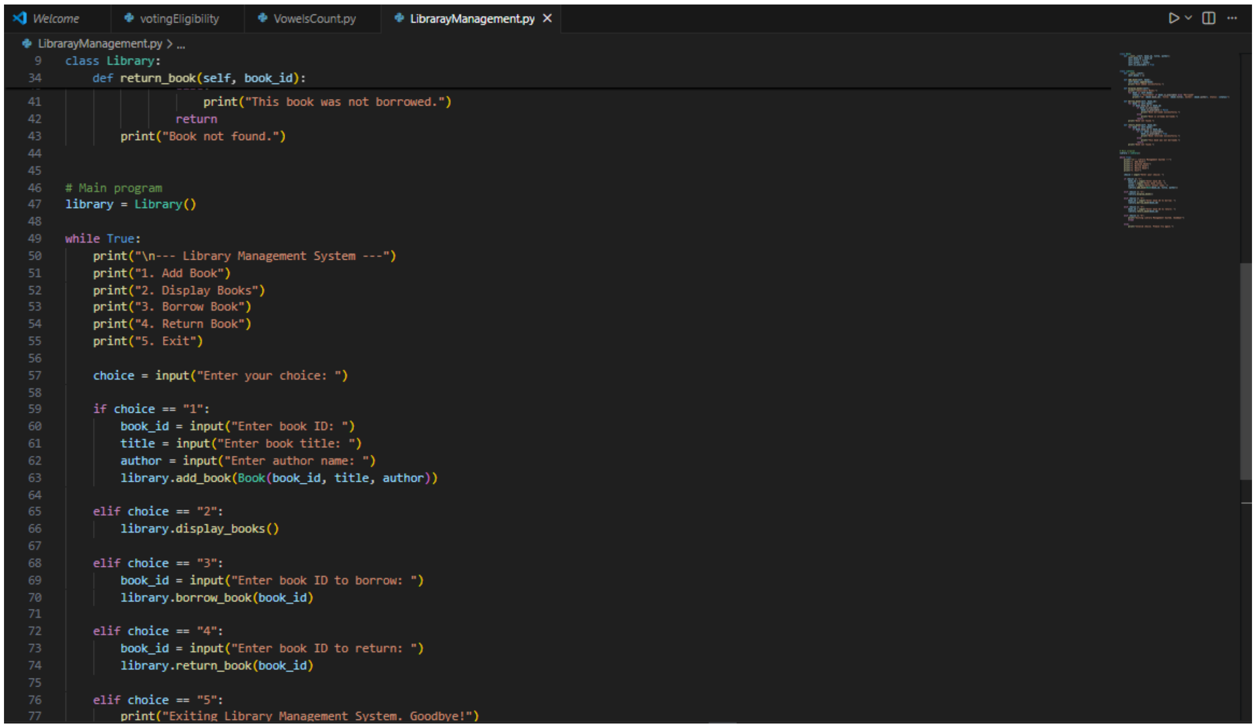
This program manages library books using a class.
Loops are used to display a menu repeatedly.
Conditional statements handle user choices.
The system allows adding and viewing books.

Algorithm:

Start
Create Library class
Display menu using loop
Perform operations based on user choice
Stop

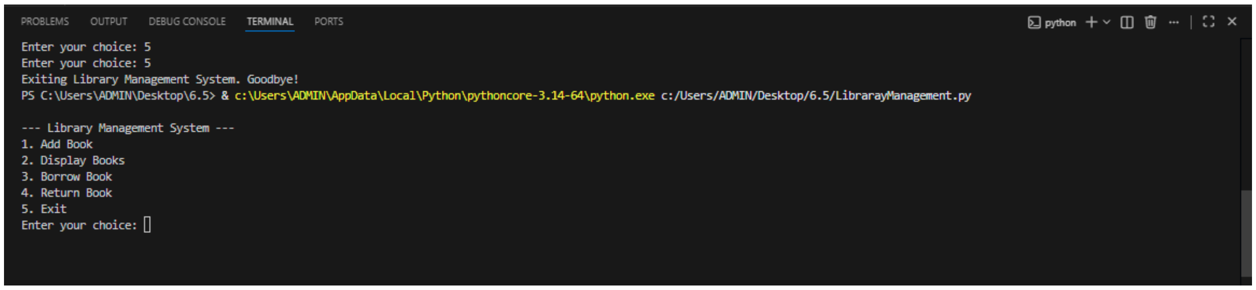


```python
class Book:
    def __init__(self, book_id, title, author):
        self.book_id = book_id
        self.title = title
        self.author = author
        self.is_available = True


class Library:
    def __init__(self):
        self.books = []

    def add_book(self, book):
        self.books.append(book)
        print("Book added successfully.")

    def display_books(self):
        print("\nAvailable Books:")
        for book in self.books:
            status = "Available" if book.is_available else "Borrowed"
            print(f"ID: {book.book_id}, Title: {book.title}, Author: {book.author}, Status: {status}")

    def borrow_book(self, book_id):
        for book in self.books:
            if book.book_id == book_id:
                if book.is_available:
                    book.is_available = False
                    print("Book borrowed successfully.")
                else:
                    print("Book is already borrowed.")
                return
        print("Book not found.")

    def return_book(self, book_id):
        for book in self.books:
            if book.book_id == book_id:
                if not book.is_available:
                    book.is_available = True
                    print("Book returned successfully.")
```



```python
class Library:
    def return_book(self, book_id):
                print("This book was not borrowed.")
            return
        print("Book not found.")


# Main program
library = Library()

while True:
    print("\n--- Library Management System ---")
    print("1. Add Book")
    print("2. Display Books")
    print("3. Borrow Book")
    print("4. Return Book")
    print("5. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        book_id = input("Enter book ID: ")
        title = input("Enter book title: ")
        author = input("Enter author name: ")
        library.add_book(Book(book_id, title, author))

    elif choice == "2":
        library.display_books()

    elif choice == "3":
        book_id = input("Enter book ID to borrow: ")
        library.borrow_book(book_id)

    elif choice == "4":
        book_id = input("Enter book ID to return: ")
        library.return_book(book_id)

    elif choice == "5":
        print("Exiting Library Management System. Goodbye!")
```

OUTPUT



```
Enter your choice: 5
Enter your choice: 5
Exiting Library Management System. Goodbye!
PS C:\Users\ADMIN\Desktop\6.5> & c:\Users\ADMIN\AppData\Local\Python\pythoncore-3.14-64\python.exe c:/Users/ADMIN/Desktop/6.5/LibrarayManagement.py

--- Library Management System ---
1. Add Book
2. Display Books
3. Borrow Book
4. Return Book
5. Exit
Enter your choice: 
```

Task 4: Attendance Management System

Explanation :

This program records student attendance using a class.

A loop is used to mark attendance for multiple students.

Conditional statements assign present or absent status.

Attendance details are displayed at the end.

Algorithm:

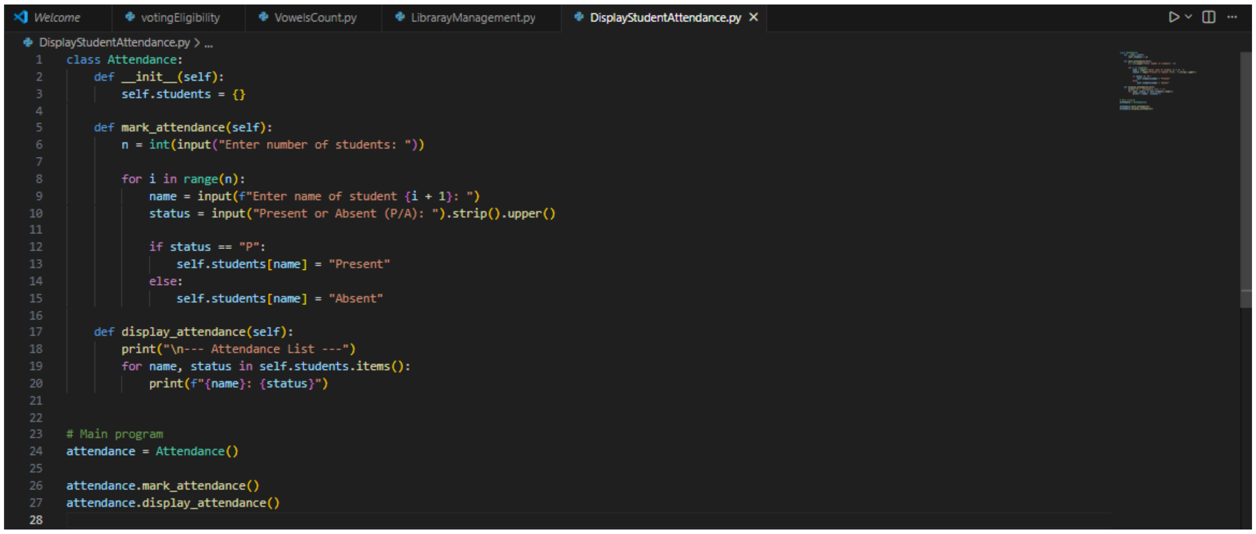
Start

Create Attendance class

Input student names and attendance using loop

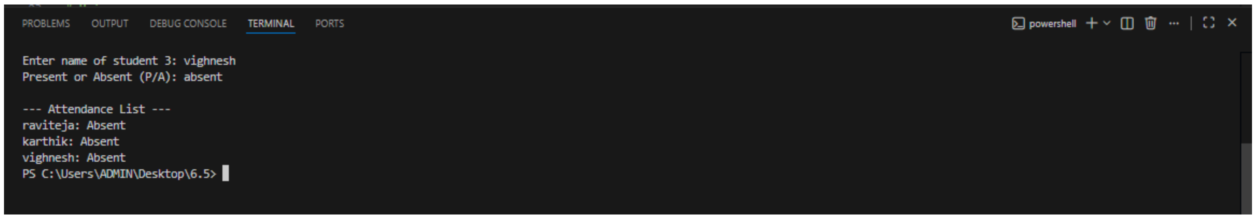
Display attendance list

Stop


Welcome
votingEligibility
VowelsCount.py
LibraryManagement.py
DisplayStudentAttendance.py
DisplayStudentAttendance.py > ...

```python
class Attendance:
    def __init__(self):
        self.students = {}

    def mark_attendance(self):
        n = int(input("Enter number of students: "))

        for i in range(n):
            name = input(f"Enter name of student {i + 1}: ")
            status = input("Present or Absent (P/A): ").strip().upper()

            if status == "P":
                self.students[name] = "Present"
            else:
                self.students[name] = "Absent"

    def display_attendance(self):
        print("\n--- Attendance List ---")
        for name, status in self.students.items():
            print(f"{name}: {status}")


# Main program
attendance = Attendance()

attendance.mark_attendance()
attendance.display_attendance()
```

OUTPUT



```
Enter name of student 3: vighnesh
Present or Absent (P/A): absent

--- Attendance List ---
raviteja: Absent
karthik: Absent
vighnesh: Absent
PS C:\Users\ADMIN\Desktop\6.5>
```

Task 5: ATM Menu Simulation

Prompt:

“Generate a Python class to mark and display student attendance using loops.”

Explanation :

This program simulates ATM operations using a menu.
A loop allows multiple transactions.
Conditional statements process user selections.
The program exits when the user chooses exit.

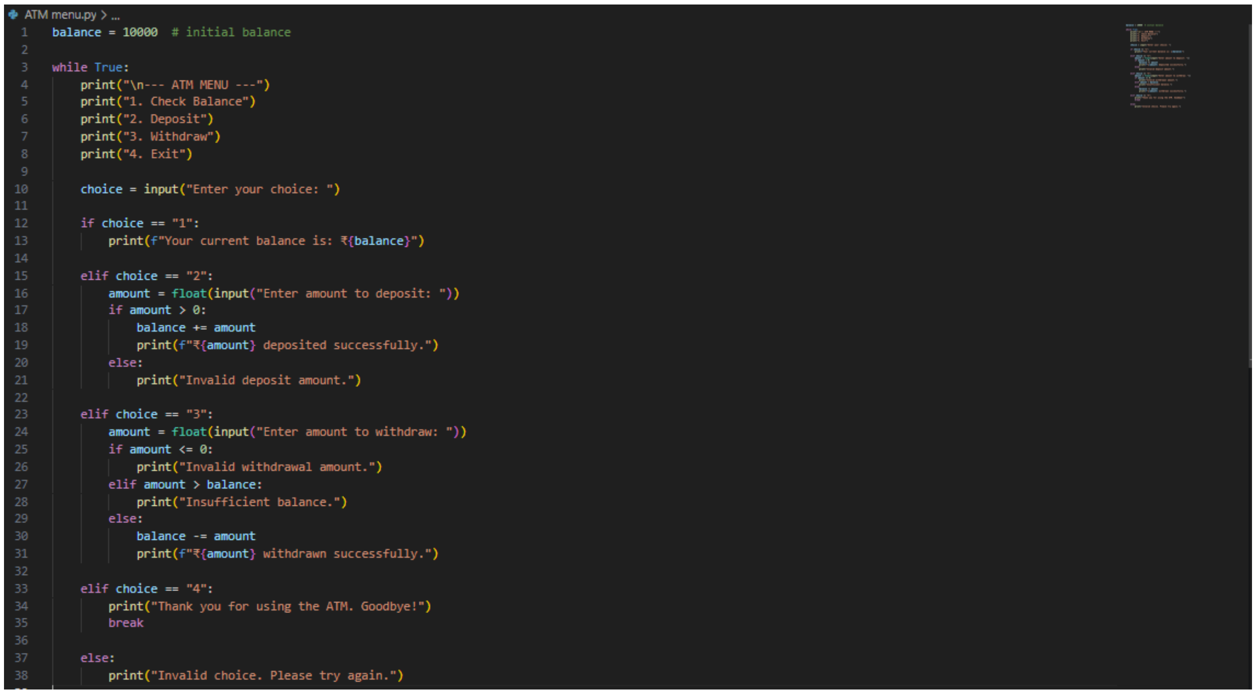
Algorithm:

Start
Initialize account balance
Display ATM menu in a loop
Perform transaction based on choice
Stop

Prompt

“Generate a Python program using loops and conditionals

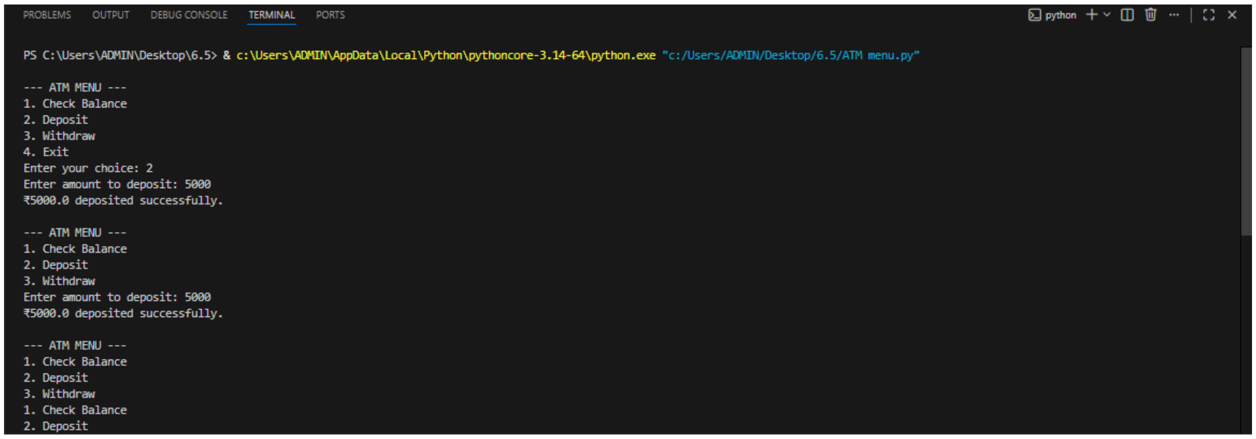
to simulate an ATM menu.”



ATM menu.py > ...

```python
balance = 10000  # initial balance

while True:
    print("\n--- ATM MENU ---")
    print("1. Check Balance")
    print("2. Deposit")
    print("3. Withdraw")
    print("4. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
        print(f"Your current balance is: ₹{balance}")

    elif choice == "2":
        amount = float(input("Enter amount to deposit: "))
        if amount > 0:
            balance += amount
            print(f"₹{amount} deposited successfully.")
        else:
            print("Invalid deposit amount.")

    elif choice == "3":
        amount = float(input("Enter amount to withdraw: "))
        if amount <= 0:
            print("Invalid withdrawal amount.")
        elif amount > balance:
            print("Insufficient balance.")
        else:
            balance -= amount
            print(f"₹{amount} withdrawn successfully.")

    elif choice == "4":
        print("Thank you for using the ATM. Goodbye!")
        break

    else:
        print("Invalid choice. Please try again.")
```

OUTPUT



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ADMIN\Desktop\6.5> & c:\Users\ADMIN\AppData\Local\Python\pythoncore-3.14-64\python.exe "c:/Users/ADMIN/Desktop/6.5/ATM menu.py"

--- ATM MENU ---
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
Enter your choice: 2
Enter amount to deposit: 5000
₹5000.0 deposited successfully.

--- ATM MENU ---
1. Check Balance
2. Deposit
3. Withdraw
Enter amount to deposit: 5000
₹5000.0 deposited successfully.

--- ATM MENU ---
1. Check Balance
2. Deposit
3. Withdraw
1. Check Balance
2. Deposit
```