**39. Program to find the sum of all prime numbers between 1 and 1000 Program that reads set of integers and displays first and second largest numbers**

# Function to check if a number is prime

def is\_prime(n):

if n < 2:

return False

for i in range(2, int(n\*\*0.5) + 1):

if n % i == 0:

return False

return True

# Find sum of all prime numbers between 1 and 1000

prime\_sum = sum(n for n in range(1, 1001) if is\_prime(n))

# Print the result

print("Sum of all prime numbers between 1 and 1000:", prime\_sum)

40). **Program to print the sum of first ‘n’ natural numbers.**

# Get user input

n = int(input("Enter a positive integer: "))

# Calculate sum using formula

sum\_n = n \* (n + 1) // 2

# Print result

print(f"Sum of first {n} natural numbers is: {sum\_n}")

ASSESMENT: Question: Bank Account Management System

Objective: You need to implement a simple Bank Account

Management System that allows the user to:

1. Check account balance.

2. Deposit money into the account.

3. Withdraw money from the account.

4. Perform transactions only if the account has enough balance

(for withdrawals).

You should use conditional statements to ensure that:

 The user can only withdraw money if the account has

sufficient balance.

 The user cannot deposit a negative amount.

class BankAccount:

def \_\_init\_\_(self, balance=0):

self.balance = balance

# Function to check account balance

def check\_balance(self):

print(f"Current balance: ${self.balance:.2f}")

# Function to deposit money

def deposit(self, amount):

if amount > 0:

self.balance += amount

print(f"Deposited: ${amount:.2f}")

else:

print("Deposit amount must be positive!")

# Function to withdraw money

def withdraw(self, amount):

if amount > self.balance:

print("Insufficient balance! Withdrawal not allowed.")

elif amount <= 0:

print("Withdrawal amount must be positive!")

else:

self.balance -= amount

print(f"Withdrawn: ${amount:.2f}")

# Create a bank account object

account = BankAccount(500) # Starting balance of $500

# Perform transactions

account.check\_balance()

account.deposit(200)

account.withdraw(100)

account.withdraw(700) # Should show insufficient balance

account.deposit(-50) # Should show an invalid deposit message

account.check\_balance()