Symbiosis Institute of Technology, Nagpur
CA-II GenAI

Sub: GenAI **Sem: VII**
Name: Vaishnavi Gulhane **PRN: 21070521088**

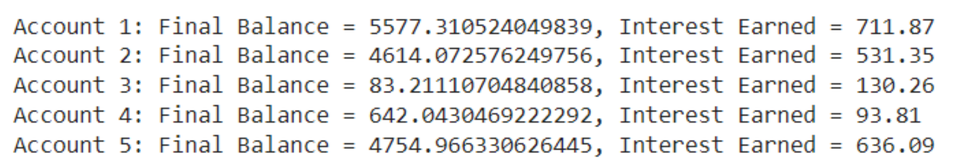
Q4 Generate a model to represent interest calculations of a Bank account where the process of calculating interest for 6 months is a. Find minimum balance for each month b. Make a total of all minimum balances c. Calculate interest based on interest rate d. Divide interest by 12 to find one-month interest e. Multiply interest by 6 to show interest in the account. Generate a model to represent transactions and interest calculations for 6 months.

Ans: the task is to model interest calculations for a bank account over 6 months based on the minimum monthly balance. Here's the step-by-step breakdown:

1. A BankAccount class is created to track the account balance and the monthly transactions (deposits and withdrawals).
2. For each of the 6 months, the program records the minimum balance. This is done by comparing the balance after each transaction in the month and keeping track of the lowest balance.
3. After collecting the minimum balances for all 6 months, the program sums these minimum balances to calculate the total minimum balance for the period.
4. Using this total, the program calculates the interest based on a given interest rate. The interest is divided by 12 to find the monthly interest.
5. Finally, the monthly interest is multiplied by 6 to get the interest for the 6-month period, and the total interest earned in the account is displayed.
6. Additionally, the program handles normal transactions (deposits and withdrawals) during the 6-month period.

```
import random
class BankAccount:
    def __init__(self, account_id, balance, interest_rate):
        self.account_id = account_id
        self.balance = balance
        self.interest_rate = interest_rate
        self.transactions = []
        self.minimum_balances = []
    def deposit(self, amount):
        self.balance += amount
        self.transactions.append(self.balance)
    def withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
        else:
            self.balance = 0
        self.transactions.append(self.balance)
    def calculate_minimum_balance(self):
        if self.transactions:
            self.minimum_balances.append(min(self.transactions))
        self.transactions = []
    def calculate_interest(self):
        total_min_balance = sum(self.minimum_balances)
        total_interest = (total_min_balance * self.interest_rate) / 12
        return total_interest * 6
    def __repr__(self):
        return f"Account {self.account_id}: Final Balance = {self.balance},
Interest Earned = {self.calculate_interest():.2f}"
def simulate_account(account, months=6):
    for month in range(months):
        for _ in range(random.randint(1, 5)):
            if random.choice([True, False]):
                account.deposit(random.uniform(50, 500))
            else:
                account.withdraw(random.uniform(50, 500))
        account.calculate_minimum_balance()
def main():
    num_accounts = 5
    interest_rate = 0.05
    accounts = []
    for i in range(1, num_accounts + 1):
        starting_balance = random.uniform(1000, 5000)
        account = BankAccount(account_id=i, balance=starting_balance,
interest_rate=interest_rate)
        simulate_account(account)
        accounts.append(account)
    for account in accounts:
        print(account)
if __name__ == "__main__":
    main()
```

Output:



```
Account 1: Final Balance = 5577.310524049839, Interest Earned = 711.87
Account 2: Final Balance = 4614.072576249756, Interest Earned = 531.35
Account 3: Final Balance = 83.21110704840858, Interest Earned = 130.26
Account 4: Final Balance = 642.0430469222292, Interest Earned = 93.81
Account 5: Final Balance = 4754.966330626445, Interest Earned = 636.09
```

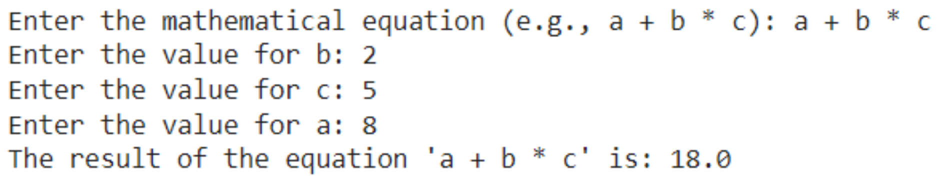
Q6 Generate a model to represent a mathematical equation, write a program to parse the equation, and ask for input for each parameter.

Ans: the task is to parse a mathematical equation and compute the result based on user input for each variable. Here's the breakdown:

1. The program accepts a mathematical equation as a string (e.g., a + b * c).
2. It identifies the variables in the equation using regular expressions.
3. The user is prompted to input values for each identified variable.
4. The program substitutes the values into the equation, replacing the variables with their respective values.
5. The final equation is evaluated using Python's eval() function to compute the result.
6. The result is then displayed to the user.

```
import re
def parse_equation(equation):
    variables = re.findall(r'[a-zA-Z]+', equation)
    return list(set(variables))
def get_variable_values(variables):
    values = {}
    for var in variables:
        values[var] = float(input(f"Enter the value for {var}: "))
    return values
def evaluate_equation(equation, values):
    for var, val in values.items():
        equation = equation.replace(var, str(val))
    return eval(equation)
def main():
    equation = input("Enter the mathematical equation (e.g., a + b * c): ")
    variables = parse_equation(equation)
    values = get_variable_values(variables)
    result = evaluate_equation(equation, values)
    print(f"The result of the equation '{equation}' is: {result}")
if __name__ == "__main__":
    main()
```

Output:



```
Enter the mathematical equation (e.g., a + b * c): a + b * c
Enter the value for b: 2
Enter the value for c: 5
Enter the value for a: 8
The result of the equation 'a + b * c' is: 18.0
```