Rajalakshmi Engineering College

Name: VIDYUT RAO s
Email: 240701587@rajalakshmi.edu.in
Roll no: 240701587
Phone: 7358733384
Branch: REC
Department: I CSE FF
Batch: 2028
Degree: B.E - CSE

Scan to verify results

NeoColab_REC_CS23221_Python Programming

REC_Python_Week 1_CY

Attempt : 2
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Shawn is planning for his younger sister's college education and wants to ensure she has enough funds when the time comes. He starts with an initial principal amount and plans to make regular monthly contributions to a savings account that offers a fixed annual interest rate.

Shawn needs to calculate the total amount that will accumulate by the time his sister is ready for college. Your task is to write a program that calculates the final amount in the savings account based on the initial principal, monthly contributions, annual interest rate, and the number of months the money is invested.

Formula:

A = P × (1 + r/n)^(n × t) + C × [((1 + r/n)^(n × t) − 1) / (r/n)]

Where:

A = Final amount after the specified time

P = Initial principal amount

C = Monthly contribution

r = Annual interest rate (as a decimal, e.g., 5% = 0.05)

n = Number of compounding periods per year (12 for monthly compounding)

t = Total time in years (months / 12)

Input Format

The first line of input consists of a float P, representing the initial principal amount.

The second line of input consists of a float R, representing the annual interest rate (in percentage).

The third line of input consists of a float C, representing the monthly contribution.

The fourth line of input consists of an integer M, representing the number of months.

Output Format

The output displays "Final amount after X months: Rs." followed by the total accumulated amount, formatted to two decimal places, where X is the number of months.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10000.0
5.0
2000.0
12

Output: Final amount after 12 months: Rs.35069.33

Answer

```
P = float(input())
R = float(input())
C = float(input())
M = int(input())

r = R / 100
n = 12
t = M / 12

amount = P * (1 + r/n)**(n*t) + C * (((1 + r/n)**(n*t) - 1) / (r/n))

print(f"Final amount after {M} months: Rs.{amount:.2f}")
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Alex is an air traffic controller who needs to record and manage flight delays efficiently. Given a flight number, the delay in minutes (as a string), and the coordinates of the flight's current position (as a complex number),

Help Alex convert and store this information in a structured format.

Input Format

The first line of input consists of an integer N, representing the flight number.

The second line consists of a string representing the delay in minutes.

The third line consists of two floats separated by a space, representing the real and imaginary parts of the complex number for the flight's position.

Output Format

The first line of output displays the complex number.

The second line displays a string with the flight number, delay, and the real and imaginary parts of the complex number, separated by commas.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 12345
30.5
12.3 45.6

Output: (12.3+45.6j)
12345, 30.5, 12.3, 45.6

Answer

```
N = int(input())
delay = input()
real, imag = map(float, input().split())

position = complex(real, imag)

print(position)
print(f"{N},{delay}, {real}, {imag}")
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

Nina is working on a project involving multiple sensors. Each sensor provides a data point that needs to be processed to compute an aggregated value.

Given data points from three sensors, write a program to calculate the aggregated value using specific bitwise operations and arithmetic manipulations. The final result should be the aggregated value modulo 1000.

Example:

Input:

1 //sensor 1 data

2 //sensor 2 data

3 //sensor 3 data

Output

9

Explanation

Calculate the bitwise AND of sensor 1 data and sensor 2 data: 0

Calculate the XOR of the result from step 1 and sensor 3 data: 3

Multiply the result from step 2 by 3: 9

Compute the final aggregated value by taking the result from step 3 modulo 1000: 9

So, the aggregated value is 9.

Input Format

The first line of input consists of an integer S1, representing sensor1 data.

The second line of input consists of an integer S2, representing sensor2 data.

The third line of input consists of an integer S3, representing sensor3 data.

Output Format

The output displays an integer representing the aggregated value.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1
2
3
Output: 9

Answer

```
S1 = int(input())
S2 = int(input())
S3 = int(input())

and_result = S1 & S2
xor_result = and_result ^ S3
aggregated_value = (xor_result * 3) % 1000

print(aggregated_value)
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Mandy is working on a mathematical research project involving complex numbers. For her calculations, she often needs to swap the real and imaginary parts of two complex numbers.

Mandy needs a Python program that takes two complex numbers as input and swaps their real and imaginary values.

Input Format

The first line of input consists of a complex number in the format a+bj, representing the first complex number.

The second line consists of a complex number in the format a+bj, representing the second complex number.

Output Format

The first line of output displays "New first complex number: " followed by the swapped complex number.

The second line of output displays "New second complex number: " followed by the swapped complex number.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10+8j
7-9j
Output: New first complex number: (8+10j)
New second complex number: (-9+7j)

Answer

```
c1 = complex(input())
c2 = complex(input())

new_c1 = complex(c1.imag, c1.real)
new_c2 = complex(c2.imag, c2.real)

print(f"New first complex number: {new_c1}")
print(f"New second complex number: {new_c2}")
```

Status : Correct ***Marks : 10/10***