

# Rajalakshmi Engineering College

Name: Vishaal S  
Email: 240701594@rajalakshmi.edu.in  
Roll no: 240701594  
Phone: 8925111576  
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 : 1  
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:

$$A = P \times (1 + r/n)^{(n \times t)} + C \times [(1 + r/n)^{(n \times t)} - 1] / (r/n)$$

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())
r=r/100
c=float(input())
m=int(input())
t=m/12
a=p*(1+r/12)**m
sip=c*(((1+r/12)**m-1)/(r/12))
tot=a+sip
print(f"Final amount after {m} months: Rs.{tot:.2f}")
```

**Status :** Correct

**Marks :** 10/10

## 2. 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**

```
a=complex(input())
b=complex(input())
c=int(a.real)
d=int(a.imag)
e=int(b.real)
f=int(b.imag)
if(c<0):
    print("New first complex number: ({}{}j)".format(d,c))
else:
    print("New first complex number: ({}+{}j)".format(d,c))
if(e<0):
    print("New second complex number: ({}{}j)".format(f,e))
else:
    print("New second complex number: ({}+{}j)".format(f,e))
```

**Status :** Correct

**Marks :** 10/10

### **3. Problem Statement**

Liam and his friends are sharing the cost of a group purchase. The total cost of the purchase is subject to a 10% discount. One of the friends receives a 35% bonus, which means they will pay a larger portion of the discounted cost. The remaining cost is then divided equally among the other friends.

Write a program to:

Calculate the total cost after applying a 10% discount. Determine the

amount paid by the friend who receives a 35% bonus. Calculate the amount each of the other friends will pay.

### ***Input Format***

The first line of input consists of a float value  $f$ , representing the total cost.

The second line contains an integer value  $n$ , representing the total number of friends.

### ***Output Format***

The first line of output displays "Cost after a 10% discount: " followed by the discounted cost of the ticket package as a float value formatted to two decimal places.

The second line displays "Friend with a 35% bonus pays: " followed by the amount paid by the friend with the bonus as a float value formatted to two decimal places.

The third line displays "Each of the other friends pays: " followed by the individual share of the remaining cost as a float value formatted to two decimal places.

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 10000.0

5

Output: Cost after a 10% discount: 9000.00

Friend with a 35% bonus pays: 3150.00

Each of the other friends pays: 1462.50

### ***Answer***

```
f=float(input())
n=int(input())
disc=(10/100)*f
f1=f-disc
high=(35/100)*f1
rem=f1-high
amount=rem/(n-1)
```

```
a='%.2f'%f1
b='%.2f'%high
c='%.2f' %amount
print(f"Cost after a 10% discount: {a}")
print(f"Friend with a 35% bonus pays: {b}")
print(f"Each of the other friends pays: {c}")
```

**Status :** Correct

**Marks :** 10/10

#### 4. 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***

```
a=int(input())
b=int(input())
c=int(input())
n=(a&b)
m=(n^c)
r=(m*3)
print(r%1000)
```

**Status :** Correct

**Marks :** 10/10