

# Rajalakshmi Engineering College

Name: Monisha S  
Email: 240701335@rajalakshmi.edu.in  
Roll no: 240701335  
Phone: 9444043820  
Branch: REC  
Department: I CSE FD  
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**

```
# You are using Python
p=float(input())
rate=float(input())
c=float(input())
m=int(input())
r=rate/100
t=m/12
n=12
amount=p*(1+r/n)**(n*t)+c*(((1+r/n)**(n*t)-1)/(r/n))
print("Final amount after ",m,f" months: Rs.{amount:.2f}")
```

**Status :** Correct

**Marks :** 10/10

## 2. 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**

# You are using Python

```
cost=float(input())
```

```
n=int(input())
```

```
d=0.1*cost
```

```
ncost=cost-d;
```

```
bonus=0.35*ncost
```

```
other=(ncost-bonus)/(n-1)
```

```
print(f"Cost after a 10% discount: {ncost:.2f}\nfriend with a 35% bonus pays: {bonus:.2f}\nEach of the other friends pays: {other:.2f}")
```

**Status : Correct**

**Marks : 10/10**

### **3. 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***

```
# You are using Python
n=int(input())
m=input()
m=float(m)
x,y=input().split()
x=float(x)
y=float(y)
z=complex(x,y)
print(z)
print(n," ",m," ",z.real," ",z.imag)
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

John is developing a financial application to help users manage their investment portfolios. As part of the application, he needs to write a program that receives the portfolio's main value and the values of two specific investments as inputs. The program should then display these values in reverse order for clear visualization.

Help John achieve this functionality by writing the required program.

##### ***Input Format***

The first line of input consists of a float, representing the first investment value.

The second line of input consists of a float, representing the second investment value.

The third line of input consists of an integer, representing the portfolio ID.

##### ***Output Format***

The first line of output prints "The values in the reverse order:".

The second line prints the integer, representing the portfolio ID.

The third line prints the second float, representing the second investment value.

The fourth line prints the first float, representing the first investment value.

Refer to the sample output for the formatting specifications.

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

Input: 35.29

9374.11

48

Output: The values in the reverse order:

48

9374.11

35.29

**Answer**

```
# You are using Python
x=float(input())
y=float(input())
z=int(input())
print("The values in the reverse order:")
print(z)
print(y)
print(x)
```

**Status :** Correct

**Marks : 10/10**