

Rajalakshmi Engineering College

Name: Akshaya Kirubakarraj
Email: 241501016@rajalakshmi.edu.in
Roll no: 241501016
Phone: 9940136549
Branch: REC
Department: I AI & ML FA
Batch: 2028
Degree: B.E - AI & ML

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

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())
dc=f*0.9
bonus_friend_share=dc*0.35
remaining_cost=dc-bonus_friend_share
other_friend_share=remaining_cost/(n-1)
print(f"Cost after a 10% discount: {dc: .2f}")
print(f"Friend with a 35% bonus pays: {bonus_friend_share: .2f}")
print(f"Each of the other friends pays: {other_friend_share: .2f}")
```

Status : Correct

Marks : 10/10

2. 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

```
first=float(input())
second=float(input())
third=int(input())
print("The values in the reverse order:")
print(third)
print(second)
print(first)
```

Status : Correct

Marks : 10/10

3. 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())
```

```
c=float(input())
```

```
m=int(input())
```

```
r=r/100
```

```
n=12
```

```
t=m/12
```

```
a1=p*(1+r/n)**(n*t)
```

```
a2=c*(((1+r/n)**(n*t)-1)/(r/n))
```

```
a= a1+a2
```

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

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

```
a=input().strip()
b=input().strip()
c1=complex(a)
c2=complex(b)
c1_swapped=f"({int(c1.imag)}){'+' if c1.real >=0 else ''}{int(c1.real)}j)"
c2_swapped=f"({int(c2.imag)}){'+' if c2.real >=0 else ''}{int(c2.real)}j)"
print("New first complex number:",c1_swapped)
print("New second complex number:",c2_swapped)
```

Status : Correct

Marks : 10/10