

# Rajalakshmi Engineering College

Name: Dineshraj R  
Email: 241501049@rajalakshmi.edu.in  
Roll no: 241501049  
Phone: 9363708090  
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

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**

```
X=float(input())
```

```
Y=float(input())
```

```
Z=int(input())
```

```
print("The values in the reverse order:")
```

```
print(Z,Y,X,sep="\n")
```

**Status :** Correct

**Marks :** 10/10

## **2. 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())
```

```
X=S1&S2
```

```
Y=X^S3
```

```
Z=Y*3
```

```
A=Z%1000
```

```
print(A)
```

**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**

```
f=int(input().strip())
d=input().strip()
real,imag=map(float,input().strip().split())
c=complex(real,imag)
print(c)
print(f"{f},{d},{real},{imag}")
```

**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=complex(input())
b=complex(input())
n=complex(a.imag,a.real)
m=complex(b.imag,b.real)
print("New first complex number:",n)
print("New second complex number:",m)
```

**Status : Correct**

**Marks : 10/10**