

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 1\_MCQ

Attempt : 1  
Total Mark : 15  
Marks Obtained : 13

#### Section 1 : MCQ

1. What is the return type of the function id?

**Answer**

int

**Status : Correct**

**Marks : 1/1**

2. What does  $3^4$  evaluate to?

**Answer**

7

**Status : Correct**

**Marks : 1/1**

3. Which of the following expressions results in an error?

**Answer**

```
int('10.8')&nbsp;
```

**Status :** Correct

**Marks :** 1/1

4. What is the output of the below expression?

```
print(3*1**3)
```

**Answer**

3

**Status :** Correct

**Marks :** 1/1

5. What is the value of the following expression?

```
float(22//3+3/3)
```

**Answer**

8.0

**Status :** Correct

**Marks :** 1/1

6. What will the following code output?

```
z = 3 + 4j  
print(abs(z))
```

**Answer**

3.0

**Status :** Wrong

**Marks :** 0/1

7. What is the output of the following program?

```
print((1, 2) + (3, 4))
```

**Answer**

error

**Status : Wrong**

**Marks : 0/1**

8. What will be the output of the following code?

```
X = 2+9*((3*12)-8)/10  
print(X)
```

**Answer**

27.2

**Status : Correct**

**Marks : 1/1**

9. What is used to concatenate two strings in Python?

**Answer**

+ operator

**Status : Correct**

**Marks : 1/1**

10. Which of the following represents the bitwise XOR operator?

**Answer**

^

**Status : Correct**

**Marks : 1/1**

11. Which of the following can convert the string to a float number?

**Answer**

float(str)

**Status : Correct**

**Marks : 1/1**

12. Which of these is not a core data type?

**Answer**

Class

**Status :** Correct

**Marks :** 1/1

13. What is the value of the following expression?

$8/4/2$ ,  $8/(4/2)$

**Answer**

(1.0,4.0)

**Status :** Correct

**Marks :** 1/1

14. Which of the following is an example of the type casting?

**Answer**

All of the above

**Status :** Correct

**Marks :** 1/1

15. The value of the expressions  $4/(3*(2-1))$  and  $4/3*(2-1)$  is the same.

True or False?

**Answer**

True

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 1\_COD

Attempt : 1  
Total Mark : 5  
Marks Obtained : 5

### Section 1 : Coding

#### 1. Problem Statement

A company has hired two employees, Alice and Bob. The company wants to swap the salaries of both employees. Alice's salary is an integer value and Bob's salary is a floating-point value.

Write a program to swap their salaries and print the new salary of each employee.

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

The first line of input consists of an integer N, representing Alice's salary.

The second line consists of a float value F, representing Bob's salary.

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

The first line of output displays "Initial salaries:"

The second line displays "Alice's salary = N", where N is Alice's salary.

The third line of output displays "Bob's salary = F", where F is Bob's salary.

After a new line space, the following line displays "New salaries after swapping:"

The next line displays "Alice's salary = X", where X is the swapped salary.

The last line displays "Bob's salary = Y", where Y is the swapped salary.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 10000

15400.55

Output: Initial salaries:

Alice's salary = 10000

Bob's salary = 15400.55

New salaries after swapping:

Alice's salary = 15400.55

Bob's salary = 10000

### **Answer**

```
# You are using Python
N=int(input())
F=float(input())
print("Initial salaries:")
print("Alice's salary =",N)
print("Bob's salary =",F)
Y=N
X=F
print("New salaries after swapping:")
print("Alice's salary =",X)
print("Bob's salary =",Y)
```

**Status : Correct**

**Marks : 1/1**

## 2. Problem Statement

A science experiment produces a decimal value as the result. However, the scientist needs to convert this value into an integer so that it can be used in further calculations.

Write a Python program that takes a floating-point number as input and converts it into an integer.

### ***Input Format***

The input consists of a floating point number, F.

### ***Output Format***

The output prints "The integer value of F is: {result}", followed by the integer number equivalent to the floating point number.

Refer to the sample output for the formatting specifications.

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

Input: 10.36

Output: The integer value of 10.36 is: 10

### ***Answer***

```
# You are using Python
F=float(input())
X=int(F)
print("The integer value of",F,"is:",X)
```

**Status :** Correct

**Marks :** 1/1

## 3. Problem Statement

In a family, two children receive allowances based on the gardening tasks they complete. The older child receives an allowance rate of Rs.5 for each task, with a base allowance of Rs.50. The younger child receives an allowance rate of Rs.3 for each task, with a base allowance of Rs.30.

Your task is to calculate and display the allowances for the older and younger children based on the number of gardening tasks they complete, along with the total allowance for both children combined.

### ***Input Format***

The first line of input consists of an integer  $n$ , representing the number of chores completed by the older child.

The second line consists of an integer  $m$ , representing the number of chores completed by the youngest child.

### ***Output Format***

The first line of output displays "Older child allowance: Rs." followed by an integer representing the allowance calculated for the older sibling.

The second line displays "Younger child allowance: Rs." followed by an integer representing the allowance calculated for the youngest sibling.

The third line displays "Total allowance: Rs." followed by an integer representing the sum of both siblings' allowances.

Refer to the sample output for formatting specifications.

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

Input: 10

5

Output: Older child allowance: Rs.100

Younger child allowance: Rs.45

Total allowance: Rs.145

### ***Answer***

```
n=int(input())
m=int(input())
A=50+(5*n)
B=30+(3*m)
print("Older child allowance:Rs.",A)
print("Younger child allowance:Rs.",B)
```



```
print("Total allowance:Rs.",A+B)
```

**Status :** Correct

**Marks :** 1/1

#### 4. Problem Statement

Quentin, a mathematics enthusiast, is exploring the properties of numbers. He believes that for any set of four consecutive integers, calculating the average of their fourth powers and then subtracting the product of the first and last numbers yields a constant value.

To validate his hypothesis, check if the result is indeed constant and display.

Example:

Input:

5

Output:

Constant value: 2064.5

Explanation:

Find the Average:

Average:  $(625 + 1296 + 2401 + 4096)/4 = 2104.5$

Now, we calculate the product of  $a$  and  $(a + 3)$ :

Product =  $5 \times (5 + 3) = 5 \times 8 = 40$

Final result:  $2104.5 - 40 = 2064.5$

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

The input consists of an integer  $a$ , representing the first of four consecutive integers.

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

The output displays "Constant value: " followed by the computed result based on Quentin's formula.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 5

Output: Constant value: 2064.5

### **Answer**

```
a=int(input())
B=((a**4)+((a+1)**4)+((a+2)**4)+((a+3)**4))/4
C=a*(a+3)
print("Constant value:",B-C)
```

**Status :** Correct

**Marks :** 1/1

## **5. Problem Statement**

Bob, the owner of a popular bakery, wants to create a special offer code for his customers. To generate the code, he plans to combine the day of the month with the number of items left in stock.

Help Bob to encode these two values into a unique offer code.

Note: Use the bitwise operator to calculate the offer code.

### **Example**

Input:

15

9

Output:

Offer code: 6

Explanation:

Given the day of the month 15th day (binary 1111) and there are 9 items left (binary 1001), the offer code is calculated as 0110 which is 6.

### ***Input Format***

The first line of input consists of an integer D, representing the day of the month.

The second line consists of an integer S, representing the number of items left in stock.

### ***Output Format***

The output displays "Offer code: " followed by an integer representing the encoded offer code.

Refer to the sample output for formatting specifications.

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

Input: 15

9

Output: Offer code: 6

### ***Answer***

```
D=int(input())
```

```
S=int(input())
```

```
X=D^S
```

```
print("Offer Code:",X)
```

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
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

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

```
a=float(input())
```

```
b=float(input())
```

```
c=int(input())
```

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

```
print(c)
```

```
print(b)
```

```
print(a)
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

Olivia is creating a wellness dashboard for her new fitness app, FitTrack.

She needs a program that can capture and display key details about a user's workout. The program should read the user's full name, the total steps they ran, the energy they expended in kilojoules, and the duration of their workout in hours. After collecting this information, the program will generate a detailed summary of the user's fitness activity.

Your task is to guide Olivia through the program.

### ***Input Format***

The first line of input consists of a string, representing the user's name.

The second line consists of an integer, representing the total steps taken.

The third line consists of a float value, representing the calories burned.

The fourth line consists of a float value, representing the workout duration in hours.

### ***Output Format***

The first line of output prints "User Name: " followed by the user's name.

The second line prints "Total Steps: " followed by the total steps.

The third line prints "Calories Burned: " followed by the calories burned, rounded off to one decimal place.

The fourth line prints "Workout Duration: X hours" where X is the workout duration, rounded off to one decimal place.

Refer to the sample output for formatting specifications.

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

Input: Alex

10000

350.5

1.5

Output: User Name: Alex

Total Steps: 10000

Calories Burned: 350.5  
Workout Duration: 1.5 hours

**Answer**

```
# You are using Python
a=input()
b=int(input())
c=float(input())
d=float(input())
print(f"User Name: {a}")
print(f"Total Steps: {b}")
print(f"Calories Burned: {c}")
print(f"Workout Duration: {d} hours")
```

**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**

```
a=float(input())
n=int(input())
d=(10/100)*a
t=a-d
b=t*(0.35)
r=(t-b)
s=n-1
g=r/s
print(f"Cost after a 10% discount: {t:.2f}")
print(f"Friend with a 35% bonus pays: {b:.2f}")
print(f"Each of the other friends pays: {g:.2f}")
```

**Status :** Correct

**Marks : 10/10**

## **4. Problem Statement**

Emily is organizing a taco party and needs to determine the total number of tacos required and the total cost. Each attendee at the party will consume 2 tacos. To ensure there are enough tacos:



If there are 10 or more attendees, Emily will need to provide an additional 5 tacos. If there are fewer than 10 attendees, Emily must ensure a minimum of 20 tacos are provided.

The cost of each taco is \$25. Write a program that calculates both the total number of tacos required and the total cost based on the number of attendees.

### ***Input Format***

The input consists of an integer  $n$ , representing the number of attendees.

### ***Output Format***

The first line prints "Number of tacos needed: " followed by an integer representing the number of tacos needed for  $n$  attendees.

The second line prints "Total cost: " followed by an integer representing the total cost.

Refer to the sample output for the formatting specifications.

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

Input: 10

Output: Number of tacos needed: 25

Total cost: 625

### ***Answer***

```
# You are using Python
```

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

```
t=n*2
```

```
if n>=10:
```

```
    t+=5
```

```
else:
```

```
    t=20
```

```
c=t*25
```

```
print("Number of tacos needed: ",t)
```

```
print("Total Cost: ",c)
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 1\_PAH

Attempt : 1  
Total Mark : 6  
Marks Obtained : 5.35

### Section 1 : Coding

#### 1. Problem Statement

A smart home system tracks the temperature and humidity of each room. Create a program that takes the room name (string), temperature (float), and humidity (float).

Display the room's climate details.

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

The first line of input consists of a string, representing the room name.

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

The third line consists of a float value, representing the humidity.

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

The first line of output prints "Room: " followed by the room name (string).

The second line prints "Temperature: " followed by the temperature (float) formatted to two decimal places.

The third line prints "Humidity: " followed by the humidity (float) formatted to two decimal places and a percentage sign (%).

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: Living Room

23.45

45.78

Output: Room: Living Room

Temperature: 23.45

Humidity: 45.78%

### **Answer**

# You are using Python

```
r=input()
```

```
temp=float(input())
```

```
h=float(input())
```

```
print(f"Room: {r}")
```

```
print(f"Temperature : {temp:.2f}")
```

```
print(f"Humidity: {h:.2f}%")
```

**Status : Correct**

**Marks : 1/1**

## **2. Problem Statement**

Liam works at a car dealership and is responsible for recording the details of cars that arrive at the showroom. To make his job easier, he wants a program that can take the car's make, model, and price, and display the information in a formatted summary.

Assist him in the program.

### ***Input Format***

The first line of input contains a string, representing the car make.

The second line contains a string, representing the car model.

The third line contains a float value, representing the car price.

### ***Output Format***

The first line of output prints "Car Make: ", followed by the car make.

The second line prints "Car Model: ", followed by the car model.

The third line prints "Price: ", followed by the car price, formatted to two decimal places.

Refer to the sample output for formatting specifications.

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

Input: Toyota

Camry

23450.75

Output: Car Make: Toyota

Car Model: Camry

Price: Rs.23450.75

### ***Answer***

```
# You are using Python
```

```
a=input()
```

```
b=input()
```

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

```
print(f"Car Make: {a}")
```

```
print(f"Car Model: {b}")
```

```
print(f"Price: Rs.{c:.2f}")
```

**Status :** Correct

**Marks :** 1/1

## **3. Problem Statement**

Mandy is debating with her friend Rachel about an interesting mathematical claim. Rachel asserts that for any positive integer  $n$ , the ratio of the sum of  $n$  and its triple to the integer itself is always 4. Mandy, intrigued by this statement, decides to validate it using logical operators and basic arithmetic.

She wants to confirm if the statement holds true for any positive integer  $n$ .

### ***Input Format***

The input consists of a positive integer  $n$ , representing the integer to be tested.

### ***Output Format***

The first line of output displays "Sum:" followed by an integer representing the calculated sum.

The second line displays "Rachel's statement is: " followed by a Boolean value indicating whether Rachel's statement is correct.

Refer to the sample output for the formatting specifications.

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

Input: 12

Output: Sum: 48

Rachel's statement is: True

### ***Answer***

```
# You are using Python
n=int(input())
add=n+(3*n)
final=add/n
print(f"Sum:{add}")
print("Rachel's statement is:",final==4)
```

**Status :** Correct

**Marks :** 1/1

## **4. Problem Statement**

Ella, an avid TV show enthusiast, is planning a binge-watching marathon for a new series. She has a specific routine: after watching a set number of episodes, she takes a short break.

She is provided with the following information:

Each episode of the series has a fixed duration of 45 minutes. After a certain number of episodes, there is a break of 15 minutes.

Ella wants to know the total time she will need to watch the entire series, including the breaks. Your task is to help Ella by calculating the total viewing time.

### ***Input Format***

The first line of input consists of an integer E, representing the total number of episodes in the series.

The second line consists of an integer B, representing the number of episodes watched before taking a break.

### ***Output Format***

The output prints an integer representing the total viewing time required to watch the entire series, including the breaks.

Refer to the sample output for formatting specifications.

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

Input: 5

2

Output: 255 minutes

### ***Answer***

# You are using Python

```
E=int(input())
```

```
B=int(input())
```

```
breaks=E//B
```

```
if E%B==0:
```

```
    breaks=-1
```

```
F=(E*45)+(breaks*15)
print(f"{F} minutes")
```

**Status :** Partially correct

**Marks :** 0.35/1

## 5. Problem Statement

Oliver is planning a movie night with his friends and wants to download a high-definition movie. He knows the file size of the movie in megabytes (MB) and his internet speed in megabits per second (Mbps). To ensure the movie is ready in time, Oliver needs to calculate the download time.

Your task is to write a program that calculates the download time and displays it in hours, minutes, and seconds.

Example

Input:

MB = 800

mbps = 40

Output:

Download Time: 0 hours, 2 minutes, and 40 seconds

Explanation:

Convert the file size to bits ( $800 \text{ MB} * 8 \text{ bits/byte} = 6400 \text{ megabits}$ ) and divide it by the download speed ( $6400 \text{ Mbps} / 40 \text{ Mbps} = 160 \text{ seconds}$ ). Now, convert the download time in seconds to hours, minutes, and seconds: 160 seconds is equal to 2 minutes and 40 seconds. So, the download time is 0 hours, 2 minutes and 40 seconds.

### ***Input Format***

The first line of input consists of an integer N, representing the file size in megabytes (MB).

The second line consists of an integer S, representing the network speed in megabits per second (mbps).



### Output Format

The output prints "Download Time: X hours, Y minutes, and Z seconds", where X, Y, and Z are integers representing the hours, minutes, and seconds respectively.

Refer to the sample output for formatting specifications.

### Sample Test Case

Input: 180

3

Output: Download Time: 0 hours, 8 minutes, and 0 seconds

### Answer

```
mb=int(input())
mbps=int(input())
megabits=mb*8
sec=megabits//mbps
```

```
hour=sec//3600
minute=(sec%3600)//60
second=(sec%3600)%60
```

```
print(f"Download Time:{hour} hours,{minute} minutes, and {second} seconds")
```

**Status :** Correct

**Marks :** 1/1

## 6. Problem Statement

Shawn, a passionate baker, is planning to bake cookies for a large party. His original recipe makes 15 cookies, with the following ingredient quantities: 2.5 cups of flour, 1 cup of sugar, and 0.5 cups of butter.

Write a program to calculate the amounts of flour, sugar, and butter needed for a different number of cookies. Provide the ingredient quantities for a specified number of cookies, maintaining the original proportions of the recipe.

### **Input Format**

The input consists of an integer n, representing the number of cookies.

### **Output Format**

The first line prints "Flour: X cups" where X represents the amount of flour required for n cookies, as a double value rounded to two decimal places.

The second line prints "Sugar: Y cups" where Y represents the amount of Sugar required for n, as a double value rounded to two decimal places.

The third line prints "Butter: Z cups" where Z represents the amount of flour required for n, as a double value rounded to two decimal places.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 15

Output: Flour: 2.50 cups

Sugar: 1.00 cups

Butter: 0.50 cups

### **Answer**

# You are using Python

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

```
f=2.5/15
```

```
s=1/15
```

```
b=0.5/15
```

```
fn=n*f
```

```
sn=n*s
```

```
bn=n*b
```

```
print(f"Flour: {fn:.2f} cups")
```

```
print(f"Sugar: {sn:.2f} cups")
```

```
print(f"Butter: {bn:.2f} cups")
```

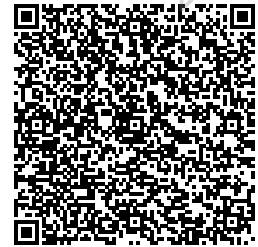
**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 2\_MCQ

Attempt : 1  
Total Mark : 15  
Marks Obtained : 13

#### Section 1 : MCQ

1. What will be the output of the following code snippet?

```
i = 0
while i < 5:
    if i % 2 == 0:
        i += 1
        continue
    print(i, end=" ")
    i += 1
```

**Answer**

1 3

**Status :** Correct

**Marks :** 1/1

2. What will be the output for the following code snippet?

```
i = 0
for i in range(10):
    break
print(i)
```

**Answer**

0

**Status :** Correct

**Marks :** 1/1

3. What will be the output of the following Python code?

```
i = 1
while True:
    if i%3 == 0:
        break
    print(i)
    i += 1
```

**Answer**

1 2

**Status :** Correct

**Marks :** 1/1

4. What will the following code output?

```
x = 0
while x < 5:
    if x == 3:
        break
    x += 1
else:
    print("Completed")
print(x)
```

**Answer**

3

**Status :** Correct

**Marks :** 1/1

5. Which keyword used in loops can skip the remaining statements for a particular iteration and start the next iteration?

**Answer**

break

**Status :** Wrong

**Marks :** 0/1

6. What is the output of the following code?

```
i = 5
while True:
    if i%009 == 0:
        break
    print(i)
    i += 1
```

**Answer**

Compile Time Error

**Status :** Correct

**Marks :** 1/1

7. What will be the output of the following Python code?

```
i = 0
while i < 5:
    print(i)
    i += 1
    if i == 3:
        break
else:
    print(0)
```

**Answer**

012

**Status :** Correct

**Marks :** 1/1

8. What is the output of the following code?

```
for i in range(5):  
    if i == 5:  
        break  
    else:  
        print(i)  
else:  
    print("Here")
```

**Answer**

0 1 2 3 4 Here

**Status :** Correct

**Marks :** 1/1

9. What will be the output of the following Python code?

```
i = 1  
while False:  
    if i%2 == 0:  
        break  
    print(i)  
    i += 2
```

**Answer**

The code runs successfully but does not print anything

**Status :** Correct

**Marks :** 1/1

10. How many times will the inner for loop be executed in the below code?

```
i=0  
while(True):  
    for j in range(4,0,-2):  
        print(i*j)  
        print("")
```

```
i=i+1  
if(i%2==0):  
    break
```

**Answer**

02

**Status :** Correct

**Marks :** 1/1

11. What will be the output of the following Python code?

```
i = 5  
while True:  
    if i%0011 == 0:  
        break  
    print(i)  
    i += 1
```

**Answer**

5 6 7 8

**Status :** Correct

**Marks :** 1/1

12. What will be the output of the following code?

```
i = 1  
while True:  
    if i%007 == 0:  
        break  
    print(i)  
    i += 1
```

**Answer**

1 2 3 4 5 6

**Status :** Correct

**Marks :** 1/1

13. When does the else statement written after the loop execute?

**Answer**

When loop condition becomes false

**Status :** Correct

**Marks :** 1/1

14. What is the output of the following program?

```
i=1
while(i<3):
    j=0
    while(j<3):
        print(i%3,end=" ")
        j=j+1
    i=i+1
```

**Answer**

0 0 0 1 1 1 2 2 2

**Status :** Wrong

**Marks :** 0/1

15. What will be the output of the following Python code?

```
i = 1
while True:
    if i % 2 == 0:
        i += 1
        continue
    if i > 10:
        break
    print(i, end = " ")
    i += 2
```

**Answer**

1 3 5 7 9

**Status :** Correct

**Marks :** 1/1



# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 2\_COD\_Updated

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

### Section 1 : Coding

#### 1. Problem Statement

Emma, a mathematics enthusiast, is exploring a range of numbers and wants to count how many of them are not Fibonacci numbers.

Help Emma determine the count of non-Fibonacci numbers within the given range [start, end] using the continue statement.

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

The first line of input consists of an integer, representing the starting number of the range.

The second line consists of an integer, representing the ending number of the range.

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

The output prints a single integer, representing the count of numbers in the range that are not Fibonacci numbers.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 1

10

Output: 5

### **Answer**

# You are using Python

```
def fib(n):
```

```
    a,b=0,1
```

```
    while a<=n:
```

```
        if a==n:
```

```
            return True
```

```
            a,b=b,a+b
```

```
    return False
```

```
start = int(input())
```

```
end=int(input())
```

```
c=0
```

```
for i in range (start,end+1):
```

```
    if fib (i):
```

```
        continue
```

```
    c+=1
```

```
print(c)
```

**Status : Correct**

**Marks : 10/10**

## **2. Problem Statement**

John, a software developer, is analyzing a sequence of numbers within a given range to calculate their digit sum. However, to simplify his task, he

excludes all numbers that are palindromes (numbers that read the same backward as forward).

Help John find the total sum of the digits of non-palindromic numbers in the range [start, end] (both inclusive).

Example:

Input:

10

20

Output:

55

Explanation:

Range [10, 20]: Non-palindromic numbers are 10, 12, 13, 14, 15, 16, 17, 18, 19 and 20.

Digit sums:  $1+0 + 1+2 + 1+3 + 1+4 + 1+5 + 1+6 + 1+7 + 1+8 + 1+9 + 2+0 = 55$ .

Output: 55

### ***Input Format***

The first line of input consists of an integer, representing the starting number of the range.

The second line of input consists of an integer, representing the ending number of the range.

### ***Output Format***

The output prints a single integer, representing the total sum of the digits of all non-palindromic numbers in the range.

Refer to the sample output for formatting specifications.

### Sample Test Case

Input: 10

20

Output: 55

### Answer

```
# You are using Python
```

```
start=int(input())
```

```
end=int (input())
```

```
total=0
```

```
for i in range(start,end+1):
```

```
    str_n = str(i)
```

```
    if(str_n!=str_n[::-1]):
```

```
        for d in str_n:
```

```
            total+=int(d)
```

```
print(total)
```

Status : Correct

Marks : 10/10

### 3. Problem Statement

Ethan, a curious mathematician, is fascinated by perfect numbers. A perfect number is a number that equals the sum of its proper divisors (excluding itself). Ethan wants to identify all perfect numbers within a given range.

Help him write a program to list these numbers.

#### Input Format

The first line of input consists of an integer start, representing the starting number of the range.

The second line consists of an integer end, representing the ending number of the range.

#### Output Format

The output prints all perfect numbers in the range, separated by a space.

Refer to the sample output for formatting specifications.

**Sample Test Case**

Input: 1

100

Output: 6 28

**Answer**

# You are using Python

```
start = int(input())
```

```
end=int(input())
```

```
P=[]
```

```
for i in range(start,end+1):
```

```
    s=0
```

```
    for j in range(1,i):
```

```
        if(i%j==0):
```

```
            s+=j
```

```
    if s==i :
```

```
        P.append(str(i))
```

```
print(" ".join(P))
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

As a junior developer working on a text analysis project, your task is to create a program that displays the consonants in a sentence provided by the user, separated by spaces.

You need to implement a program that takes a sentence as input and prints the consonants while skipping vowels and non-alphabetic characters using only control statements.

**Input Format**

The input consists of a string representing the sentence.

**Output Format**

The output displays space-separated consonants present in the sentence.

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: Hello World!

Output: H l l W r l d

**Answer**

```
# You are using Python
a=input()
V="A E I O U a e i o u"
for i in a:
    if i.isalpha() and i not in V:
        print(i,end=" ")
```

**Status :** Correct

**Marks : 10/10**

**5. Problem Statement**

You work as an instructor at a math enrichment program, and your goal is to develop a program that showcases the concept of using control statements to manipulate loops. Your task is to create a program that takes an integer 'n' as input and prints the squares of even numbers from 1 to 'n', while skipping odd numbers.

**Input Format**

The input consists of a single integer, which represents the upper limit of the range.

**Output Format**

The output displays the square of even numbers from 1 to 'n' separated by lines.

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: 10

Output: 4

16

36

64

100

**Answer**

# You are using Python

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

```
for i in range(2,n+1,2):
```

```
    print(i**2)
```

**Status :** Correct

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 2\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 37.5

### Section 1 : Coding

#### 1. Problem Statement

Nisha is a mathematics enthusiast, eager to explore the realm of twin prime numbers. The objective is to develop a program that enables the discovery and presentation of twin prime pairs.

The program should take an integer 'n' as input and generate 'n' pairs of twin primes, displaying the pairs with a difference of 2 between them.

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

The input consists of a single integer, n.

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

The output displays the 'n' pairs of twin primes, the pairs with a difference of 2 between them.



Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 5

Output: 3 5

5 7

11 13

17 19

29 31

### **Answer**

```
# You are using Python
def prime(n):
    if n<2:return False
    for i in range(2,int(n**0.5)+1):
        if n%i==0: return False
    return True

n=int(input())
count=0
num=2
while count<n:
    if prime(num) and prime(num+2):
        print(num,num+2)
        count+=1
    num+=1
```

**Status :** Correct

**Marks : 10/10**

## **2. Problem Statement**

Rohith is a data analyst who needs to categorize countries based on their population growth rates. Each country is assigned a unique code. Rohith will receive a code and corresponding data based on the code. If the data

falls within specific thresholds, he needs to classify the country's priority level.

Your task is to write a program that reads a country code and its associated data, and then determines if the priority is "High" or "Low."

Thresholds: France: Priority is "High" if the percentage < 50, else "Low". Japan: Priority is "High" if life expectancy > 80, else "Low". Brazil: Priority is "High" if the urban population > 80, else "Low".

### ***Input Format***

The first line of input consists of an integer, representing the country code (1 for France, 2 for Japan, 3 for Brazil).

If the country code is 1,

- The second line consists of a floating-point value N, representing the percentage of the English-speaking population.

If the country code is 2,

- The second line consists of a floating-point value A, representing the average life expectancy in years.

If the country code is 3,

- The second line consists of a floating-point value P, representing the percentage of the urban population.

### ***Output Format***

The first line of output displays "Priority: High" or "Priority: Low" based on the input data.

If the country code is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

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

Input: 1  
30.0

Output: Priority: High

**Answer**

```
# You are using Python
n=int(input())
a=float(input())
if n==1:
    if a<50:
        print("Priority:High")
    else:
        print("Priority:Low")
elif n==2:
    if a>80:
        print("Priority:High")
    else:
        print("Priority:Low")
elif n == 3:
    if a>80:
        print("Priority:High")
    else:
        print("Priority:Low")
else:
    print("Invalid")
```

**Status :** Partially correct

**Marks :** 7.5/10

### 3. Problem Statement

Students are allowed to work on our computer center machines only after entering the correct secret code. If the code is correct, the message "Logged In" is displayed. They are not allowed to log in to the machine until they enter the correct secret code.

Write a program to allow the student to work only if he/she enters the correct secret code.

Note: Here, secret code means the last three digits should be divisible by the first digit of the number.

### **Input Format**

The input consists of an integer n, which represents the secret code.

### **Output Format**

The output displays either "Logged In" or "Incorrect code" based on the given condition.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 2345

Output: Incorrect code

### **Answer**

```
# You are using Python
n=int(input())
a=str(n)
b=int(a[0])
c=int (a[-3:])
if c%b==0:
    print("Logged In")
else:
    print("Incorrect code")
```

**Status : Correct**

**Marks : 10/10**

## **4. Problem Statement**

Gabriel is working on a wildlife research project where he needs to compute various metrics for different animals based on their characteristics. Each animal type requires a different calculation: a deer's distance traveled, a bear's weight based on footprint size, or a bird's altitude based on its flying pattern.

Conditions:

For Deer (Mode 'D' or 'd'): Distance = speed of sound \* time taken, where the speed of sound in air is 343 meters per second. For Bear (Mode 'B' or 'b'): Weight = footprint size \* average weight, where the average weight per square inch for a bear is 5.0 pounds. For Bird (Mode 'F' or 'f'): Altitude = flying pattern \* distance covered (in meters).

Write a program to help Gabriel analyze the characteristics of animals based on the given inputs.

### ***Input Format***

The first line of input consists of a character, representing the type of animal 'D/d' for deer, 'B/b' for bear, and 'F/f' for bird.

If the choice is 'D' or 'd':

The second line of input consists of a floating-point value T, representing the time taken from the deer's location to the observer.

If the choice is 'B' or 'b':

The second line of input consists of a floating-point value S, representing the size of the bear's footprint in square inches.

If the choice is 'F' or 'f':

1. The second line of input consists of a floating-point value P, representing the bird's flying pattern.
2. The third line consists of a floating-point value D, representing the distance covered by the bird in meters.

### ***Output Format***

The output prints one of the following:

If the choice is 'D' or 'd':

The output prints "Distance: X m" where X is a floating point value rounded off to two decimal places, representing the calculated distance traveled by the sound wave in meters.

If the choice is 'B' or 'b':

The output prints "Weight: Y lb" where Y is a floating point value rounded off to two decimal places, representing the estimated weight of the bear in pounds.

If the choice is 'F' or 'f':

The output prints "Altitude: Z m" where Z is a floating point value rounded off to two decimal places, representing the calculated altitude of the bird's flight in meters.

If the given choice is invalid, print "Invalid".

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: d

2.5

Output: Distance: 857.50 m

### **Answer**

```
# You are using Python
```

```
a=input().strip()
```

```
s=343
```

```
b=5.0
```

```
if a=='D' or a=='d':
```

```
    T=float(input())
```

```
    d=s*T
```

```
    print(f"Distance :{d:.2f}m")
```

```
elif a=='B' or a=='b':
```

```
    s=float(input())
```

```
    w=s*b
```

```
    print(f"weight : {w:.2f}lb")
```

```
elif a=="F" or a=="f":
```

```
    P=float(input())
```

```
    D=float(input())
```

```
    j=P*D
```

```
    print(f"Altitude :{j:.2f}m")
```

```
else:
```

```
    print("Invalid")
```

Status : Correct

Marks : 10/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 2\_PAH\_Updated

Attempt : 1  
Total Mark : 60  
Marks Obtained : 60

### Section 1 : Coding

#### 1. Problem Statement

Imagine being entrusted with the responsibility of creating a program that simulates a math workshop for students. Your task is to develop an interactive program that not only calculates but also showcases the charm of factorial values. Your program should efficiently compute and present the sum of digits for factorial values of only odd numbers within a designated range. This approach will ingeniously keep even factorials at bay, allowing students to delve into the intriguing world of mathematics with enthusiasm and clarity.

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

The input consists of a single integer, n.

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



The output displays the factorial and sum of digits of the factorial of odd numbers within the given range.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 6

Output: 1! = 1, sum of digits = 1

3! = 6, sum of digits = 6

5! = 120, sum of digits = 3

### **Answer**

```
# You are using Python
import math
n=int(input())
for i in range (1,n+1,2):
    f=1
    for j in range(1,i+1):
        f*=j
    d=0
    for a in str(f):
        d+=int(a)
    print(i,"! =",f," sum of digits =",d)
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

As a software engineer, your goal is to develop a program that facilitates the identification of leap years in a specified range. Your task is to create a program that takes two integer inputs, representing the start and end years of the range and then prints all the leap years within that range.

### **Input Format**

The first line of the input consists of an integer, which represents the start year.

The second line consists of an integer, which represents the end year.

### **Output Format**

The output displays the leap years within the given range, separated by lines.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 2000

2053

Output: 2000

2004

2008

2012

2016

2020

2024

2028

2032

2036

2040

2044

2048

2052

### **Answer**

# You are using Python

```
a=int(input())
```

```
b=int(input())
```

```
for i in range(a,b+1):
```

```
    if((i%4==0 and i%100!=0) or i%400==0):
```

```
        print(i)
```

**Status :** Correct

**Marks : 10/10**

### **3. Problem Statement**

Aarav is fascinated by the concept of summing numbers separately based on their properties. He plans to write a program that calculates the sum of

even numbers and odd numbers separately from 1 to a given positive integer.

Aarav wants to input an integer value to represent the upper limit of the range. Help Aarav by developing a program that computes and displays the sum of even and odd numbers separately.

### ***Input Format***

The input consists of a single integer N, where N is the upper limit of the range.

### ***Output Format***

The output consists of two lines:

- The first line displays the sum of even numbers from 1 to N.
- The second line displays the sum of odd numbers from 1 to N.

Refer to the sample output for the exact format.

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

Input: 10

Output: Sum of even numbers from 1 to 10 is 30

Sum of odd numbers from 1 to 10 is 25

### ***Answer***

```
# You are using Python
n=int(input())
even,odd=0,0
for i in range (1,n+1):
    if i%2==0:
        even+=i
    else:
        odd+=i
print("Sum of even numbers from 1 to ",n,"is",even)
print("Sum of odd numbers from 1 to ",n,"is",odd)
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

Kamali recently received her electricity bill and wants to calculate the amount she needs to pay based on her usage. The electricity company charges different rates based on the number of units consumed.

For the first 100 units, there is no charge. For units consumed beyond 100 and up to 200, there is a charge of Rs. 5 per unit. For units consumed beyond 200, there is a charge of Rs. 10 per unit.

Write a program to help Kamali calculate the amount she needs to pay for her electricity bill based on the units consumed.

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

The input consists of an integer, representing the number of units.

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

The output prints the total amount of the electricity bill, an integer indicating the amount Kamali needs to pay in the format "Rs. amount".

Refer to the sample output for the exact format.

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

Input: 350

Output: Rs. 2000

##### ***Answer***

```
# You are using Python
n=int(input())
if n<=100:
    print("Rs. 0")
elif n>100 and n<=200:
    print("Rs.",(5*(n-100)))
else:
    print("Rs.",(5*100+10*(n-200)))
```

**Status :** Correct

**Marks :** 10/10

## 5. Problem Statement

Sophia, a primary school teacher, wants to calculate the sum of numbers within a given range, excluding those that are multiples of 3.

Write a program to help Sophia compute the sum of all numbers between start and end (inclusive) that are not divisible by 3 using the continue statement.

### ***Input Format***

The first line of input consists of an integer, representing the starting number of the range.

The second line of input consists of an integer, representing the ending number of the range.

### ***Output Format***

The output prints a single integer, representing the sum of numbers in the range that are not multiples of 3.

Refer to the sample output for formatting specifications.

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

Input: 1

10

Output: 37

### ***Answer***

```
# You are using Python
```

```
s=int(input())
```

```
e=int(input())
```

```
sum=0
```

```
for i in range(s,e+1):
```

```
    if i% 3!=0:
```

```
        sum+=i
```

```
print(sum)
```

Status : Correct

Marks : 10/10

## 6. Problem Statement

Rajesh wants to design a program that simulates a real-time scenario based on a mathematical concept known as the Collatz Conjecture. This concept involves the repeated application of rules to a given starting number until the number becomes 1. The rules are as follows:

If the number is even, divide it by 2. If the number is odd, multiply it by 3 and add 1.

Your task is to write a program that takes a positive integer as input, applies the Collatz Conjecture rules to it, counts the number of steps taken to reach 1, and provides an output accordingly. If the process exceeds 100 steps, the program should print a message indicating so and use break to exit.

### **Input Format**

The input consists of a single integer, n.

### **Output Format**

The output displays the total number of steps taken to reach 1 if it's under 100.

If it's more than 100, it displays "Exceeded 100 steps. Exiting...".

Refer to sample output for the formatting specifications.

### **Sample Test Case**

Input: 6

Output: Steps taken to reach 1: 8

### **Answer**

```
# You are using Python
def c(n):
    s=0
```

```
while n!=1:
    if s>=100:
        print("Exceeded 100 steps. Exiting...")
        break
    if(n%2==0):
        n//=2
    else:
        n=3*n+1
    s+=1
if s<100:
    print("Steps Taken to reach 1:",s)
n=int(input())
c(n)
```

**Status :** Correct

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 3\_MCQ

Attempt : 1  
Total Mark : 25  
Marks Obtained : 25

#### Section 1 : MCQ

1. What is the output of the following code?

```
my_list = [3, 6, 1, 2, 5, 4]  
print(sorted(my_list) == my_list.sort())
```

**Answer**

False

**Status : Correct**

**Marks : 1/1**

2. What does the append() method do in Python?

**Answer**

Adds a new element to the end of the list

**Status : Correct**

**Marks : 1/1**



3. What will be the output of the following program?

```
numbers = [1, 2, 3, 4, 5]  
numbers.append(6, 7)  
print(numbers)
```

**Answer**

Compile Time Error

**Status :** Correct

**Marks :** 1/1

4. Which method is used to add multiple items to the end of a list?

**Answer**

extend()

**Status :** Correct

**Marks :** 1/1

5. What is the output of the following Python code?

```
text = " Python "  
answer = text.strip()  
print(answer)
```

**Answer**

Python

**Status :** Correct

**Marks :** 1/1

6. Suppose list1 is [4, 2, 2, 4, 5, 2, 1, 0], Which of the following is the correct syntax for slicing operation?

**Answer**

all of the mentioned options

**Status :** Correct

**Marks :** 1/1

7. Which of the following is a valid way to use the '%' operator to

concatenate strings in Python?

**Answer**

```
"%s %s" % (string1, string2)
```

**Status : Correct**

**Marks : 1/1**

8. What will be the output of the following code?

```
my_list = [1, 2, 2, 3]  
print(my_list.count(2))
```

**Answer**

2

**Status : Correct**

**Marks : 1/1**

9. What is the output of the following Python code?

```
string1 = "Hello"  
string2 = "World"  
result = string1 + string2  
print(result)
```

**Answer**

HelloWorld

**Status : Correct**

**Marks : 1/1**

10. What does negative indexing in Python lists allow you to do?

**Answer**

Access elements in the list from the end

**Status : Correct**

**Marks : 1/1**

11. What is the result of the slicing operation `lst[-5:-2]` on the list `lst = [1, 2, 3, 4, 5, 6]`?

**Answer**

[2, 3, 4]

**Status :** Correct

**Marks :** 1/1

12. Which method in Python is used to create an empty list?

**Answer**

list()

**Status :** Correct

**Marks :** 1/1

13. What is the output of the following Python code?

```
word = "programming"  
answer = word.index("gram")  
print(answer)
```

**Answer**

3

**Status :** Correct

**Marks :** 1/1

14. Suppose list1 is [2, 33, 222, 14, 25], What is list1[:-1]?

**Answer**

[2, 33, 222, 14]

**Status :** Correct

**Marks :** 1/1

15. What does the following code output?

```
lst = [10, 20, 30, 40, 50]  
print(lst[-4:-1])
```

**Answer**

[20, 30, 40]

**Status :** Correct

**Marks :** 1/1

16. What is the output of the following Python code?

```
name = "John"  
age = 25  
message = "My name is %s and I am %d years old." % (name, age)  
print(message)
```

**Answer**

My name is John and I am 25 years old.

**Status :** Correct

**Marks :** 1/1

17. What is the output of the following Python code?

```
text = "Python"  
result = text.center(10, "*")  
print(result)
```

**Answer**

**\*\*Python\*\***

**Status :** Correct

**Marks :** 1/1

18. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1]?

**Answer**

25

**Status :** Correct

**Marks :** 1/1

19. What will be the output of the following code?

```
numbers = [1, 2, 3, 4, 5]  
numbers.remove(6)  
print(numbers)
```

**Answer**

ValueError: list.remove(x): x not in list

**Status :** Correct

**Marks :** 1/1

20. What is the output of the following Python code?

```
word = "Python"  
result = word[::-1]  
print(result)
```

**Answer**

nohtyP

**Status :** Correct

**Marks :** 1/1

21. If you have a list lst = [1, 2, 3, 4, 5, 6], what does the slicing operation lst[-3:] return?

**Answer**

The last three elements of the list

**Status :** Correct

**Marks :** 1/1

22. What is the output of the following code?

```
my_list = [1, 2, 3]  
my_list *= 2  
print(len(my_list))
```

**Answer**

6

**Status :** Correct

**Marks :** 1/1

23. What is the output of the following Python code?

```
txt = "My Classroom"
```

```
print(txt.find("o"))  
print(txt.index("o"))
```

**Answer**

99

**Status :** Correct

**Marks :** 1/1

24. What is the output of the following Python code?

```
b = "Projects!"  
print(b[2:5])
```

**Answer**

oje

**Status :** Correct

**Marks :** 1/1

25. What is the output of the following Python code?

```
a = "Hello"  
b = "World"  
c = a + " " + b  
print(c)
```

**Answer**

Hello World

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 3\_COD

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

### Section 1 : Coding

#### 1. Problem Statement

Dhruv wants to write a program to slice a given string based on user-defined start and end positions.

The program should check whether the provided positions are valid and then return the sliced portion of the string if the positions are within the string's length.

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

The first line consists of the input string as a string.

The second line consists of the start position (0-based index) as an integer.

The third line consists of the end position (0-based index) as an integer.

### **Output Format**

The output displays the following format:

If the start and end positions are valid, print the sliced string.

If the start and end positions are invalid, print "Invalid start and end positions".

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: pythonprogramming

0

5

Output: python

### **Answer**

```
# You are using Python
```

```
a=input()
```

```
s=int(input())
```

```
e=int(input())
```

```
l=len(a)
```

```
if s<e and e<l:
```

```
    print(a[s:e+1])
```

```
else:
```

```
    print("Invalid start and end positions")
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

You have a string containing a phone number in the format "(XXX) XXX-XXXX". You need to extract the area code from the phone number and create a new string that contains only the area code.

Write a Python program for the same.

Note



(XXX) - Area code

XXX-XXXX - Phone number

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

The input consists of a string, representing the phone number in the format "(XXX) XXX-XXXX".

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

The output displays "Area code: " followed by a string representing the area code for the given phone number.

Refer to the sample output for the formatting specifications.

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

Input: (123) 456-7890

Output: Area code: 123

#### ***Answer***

```
# You are using Python
a=input()
print("Area code:",a[1:4])
```

**Status :** Correct

**Marks :** 10/10

### **3. Problem Statement**

Ram is working on a program to manipulate strings. He wants to create a program that takes two strings as input, reverses the second string, and then concatenates it with the first string.

Ram needs your help to design a program.

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

The input consists of two strings in separate lines.

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

The output displays a single line containing the concatenated string of the first string and the reversed second string.

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: hello  
word

Output: hellodrow

**Answer**

```
# You are using Python
a=input()
b=input()
c=a+b[::-1]
print(c)
```

**Status :** Correct

**Marks :** 10/10

**4. Problem Statement**

Alex is working on a Python program to manage a list of elements. He needs to append multiple elements to the list and then remove an element from the list at a specified index.

Your task is to create a program that helps Alex manage the list. The program should allow Alex to input a list of elements, append them to the existing list, and then remove an element at a specified index.

**Input Format**

The first line contains an integer  $n$ , representing the number of elements to be appended to the list.

The next  $n$  lines contain integers, representing the elements to be appended to the list.

The third line of input consists of an integer M, representing the index of the element to be popped from the list.

### **Output Format**

The first line of output displays the original list.

The second line of output displays the list after popping the element of the index M.

The third line of output displays the popped element.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 5

64

98

-1

5

26

3

Output: List after appending elements: [64, 98, -1, 5, 26]

List after popping last element: [64, 98, -1, 26]

Popped element: 5

### **Answer**

```
# You are using Python
```

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

```
a=[]
```

```
for i in range (n):
```

```
    b=int(input())
```

```
    a.append(b)
```

```
print("List after appending elements:",a)
```

```
c=int(input())
```

```
d=a.pop(c)
```

```
print("List after popping last element:",a)
```

```
print("Popped element:",d)
```

**Status : Correct**

**Marks : 10/10**

## 5. Problem Statement

Given a list of positive and negative numbers, arrange them such that all negative integers appear before all the positive integers in the array. The order of appearance should be maintained.

Example

Input:

[12, 11, -13, -5, 6, -7, 5, -3, -6]

Output:

List = [-13, -5, -7, -3, -6, 12, 11, 6, 5]

Explanation:

The output is the arranged list where all the negative integers appear before the positive integers while maintaining the original order of appearance.

### ***Input Format***

The input consists of a single line containing a list of integers enclosed in square brackets separated by commas.

### ***Output Format***

The output displays "List =" followed by an arranged list of integers as required, separated by commas and enclosed in square brackets.

Refer to the sample output for the formatting specifications.

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

Input: [12, 11, -13, -5, 6, -7, 5, -3, -6]

Output: List = [-13, -5, -7, -3, -6, 12, 11, 6, 5]

### ***Answer***

# You are using Python

```
a=eval(input())
n=len(a)
c=[]
d=list()
for i in range(n):
    if a[i]<0:
        c.append(a[i])
    else:
        d.append(a[i])
print("List=",c+d)
```

**Status :** Correct

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 3\_PAH

Attempt : 1  
Total Mark : 60  
Marks Obtained : 60

### Section 1 : Coding

#### 1. Problem Statement

Accept an unsorted list of length  $n$  with both positive and negative integers, including 0. The task is to find the smallest positive number missing from the array. Assume the  $n$  value is always greater than zero.

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

The first line consists of  $n$ , which means the number of elements in the array.

The second line consists of the values in the list as space-separated integers.

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

The output displays the smallest positive number, which is missing from the array.

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: 6  
-5 2 0 -1 -10 2

Output: 1

**Answer**

```
# You are using Python
n=int(input())
a=list(map(int,input().split()))
a.sort()
i=1
for j in a:
    if j==i:
        i+=1
    elif j>i:
        break
print(i)
```

**Status :** Correct

**Marks : 10/10**

**2. Problem Statement**

Neha is learning string operations in Python and wants to practice using built-in functions. She is given a string A, and her task is to:

Find the length of the string using a built-in function. Copy the content of A into another string B using built-in functionality.

Help Neha implement a program that efficiently performs these operations.

**Input Format**

The input consists of a single line containing the string A (without spaces).

**Output Format**

The first line of output prints the length of the given string.

The second line prints the copied string without an extra newline at the end.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: technology-23

Output: Length of the string: 13

Copied string: technology-23

### **Answer**

# You are using Python

```
a=input()
```

```
b=str(a)
```

```
print("Length of the string:",len(a))
```

```
print("Copied string:",b)
```

**Status :** Correct

**Marks : 10/10**

## **3. Problem Statement**

Kyara is analyzing a series of measurements taken over time. She needs to identify all the "peaks" in this list of integers.

A peak is defined as an element that is greater than its immediate neighbors. Boundary elements are considered peaks if they are greater than their single neighbor.

Your task is to find and list all such peaks using list comprehension.

### **Example**

Input

1 3 2 4 1 5 7 6 10 2 8

Output



Peaks: [3, 4, 7, 10, 8]

#### Explanation

3 is a peak because it's greater than 1 and 2.

4 is a peak because it's greater than 2 and 1.

7 is a peak because it's greater than 5 and 6.

10 is a peak because it's greater than 6 and 2.

8 is a peak because it is an boundary element and it is greater than 2.

#### Input Format

The input consists of several integers separated by spaces, representing the measurements.

#### Output Format

The output displays "Peaks: " followed by a list of integers, representing the peak elements in the list.

Refer to the sample output for the formatting specifications.

#### Sample Test Case

Input: 1 3 2 4 1 5 7 6 10 2 8

Output: Peaks: [3, 4, 7, 10, 8]

#### Answer

```
# You are using Python
a=list(map(int,input().split()))
b=[]
n=len(a)
for i in range(n):
    p=False
    if (i==0 and n>1 and a[i]>a[i+1]):
        p=True
    elif (i==n-1 and n>1 and a[i]>a[i-1]):
        p=True
    elif(0<i<n-1 and a[i]>a[i-1] and a[i]>a[i+1]):
```

```
p=True
elif n==1:
    p=True
if p:
    b.append(a[i])
print("Peaks:",b)
```

**Status :** Correct

**Marks : 10/10**

#### 4. Problem Statement

Gowri was doing her homework. She needed to write a paragraph about modern history. During that time, she noticed that some words were repeated repeatedly. She started counting the number of times a particular word was repeated.

Your task is to help Gowri to write a program to get a string from the user. Count the number of times a word is repeated in the string.

Note: Case-sensitive

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

The first line of input consists of a string, str1.

The second line consists of a single word that needs to be counted, str2.

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

The output displays the number of times the given word is in the string.

If the second string str2 is not present in the first string str1, it prints 0.

Refer to the sample output for the formatting specifications.

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

Input: I felt happy because I saw the others were happy and because I knew I should feel happy

happy

Output: 3

**Answer**

```
# You are using Python
a=input()
b=input()
print(a.count(b))
```

**Status :** Correct

**Marks : 10/10**

## 5. Problem Statement

You are tasked with writing a program that takes  $n$  integers as input from the user and stores them in a list. After this, you need to transform the list according to the following rules:

The element at index 0 should be replaced with 0. For elements at even indices (excluding index 0), replace the element with its cube. For elements at odd indices, replace the element with its square.

Additionally, you should sort the list in ascending order before applying these transformations.

### **Input Format**

The first line of input represents the size of the list,  $N$ .

The elements of the list are represented by the next  $N$  lines.

### **Output Format**

The first line of output displays "Original List: " followed by the original list.

The second line displays "Replaced List: " followed by the replacement list as per the given condition.

Refer to the sample output for the formatting specifications.

### Sample Test Case

Input: 5

5  
1  
2  
3  
4

Output: Original List: [1, 2, 3, 4, 5]

Replaced List: [0, 4, 27, 16, 125]

### Answer

# You are using Python

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

```
a=[]
```

```
for i in range(n):
```

```
    b=int(input())
```

```
    a.append(b)
```

```
s=sorted(a)
```

```
print("Original List:",s)
```

```
c=[]
```

```
for j in range(n):
```

```
    if j==0:
```

```
        c.append(0)
```

```
    elif j%2==0:
```

```
        c.append(s[j]**3)
```

```
    else:
```

```
        c.append(s[j]**2)
```

```
print("Replaced List:",c)
```

Status : Correct

Marks : 10/10

## 6. Problem Statement

Imagine you are developing a text analysis tool for a cybersecurity company. Your task is to analyze input strings to categorize and count the characters into four categories: uppercase letters, lowercase letters, digits, and special characters. The company needs this tool to process log files and identify potential security threats.

### Input Format

The input consists of the log entry provided as a single string.

### **Output Format**

The output consists of four lines:

The first line contains an integer representing the count of uppercase letters in the format "Uppercase letters: {uppercase count}".

The second line contains an integer representing the count of lowercase letters in the format "Lowercase letters: {lowercase count}".

The third line contains an integer representing the count of digits in the format "Digits: {digits count}".

The fourth line contains an integer representing the count of special characters in the format "Special characters: {special characters count}".

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: Hello123

Output: Uppercase letters: 1

Lowercase letters: 4

Digits: 3

Special characters: 0

### **Answer**

```
# You are using Python
```

```
a=input()
```

```
u,l,d,s=0,0,0,0
```

```
for i in a:
```

```
    if i.isupper():
```

```
        u+=1
```

```
    elif i.islower():
```

```
        l+=1
```

```
    elif i.isdigit():
```

```
        d+=1
```

```
    else:
```

```
        s+=1
```

```
print("Uppercase letters:",u)
print("Lowercase letters:",l)
print("Digits:",d)
print("Special characters:",s)
```

**Status :** Correct

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 3\_CY

Attempt : 1  
Total Mark : 30  
Marks Obtained : 26.5

### Section 1 : Coding

#### 1. Problem Statement

Emily is a data analyst working for a company that collects feedback from customers in the form of text messages. As part of her data validation tasks, Emily needs to perform two operations on each message:

Calculate the sum of all the digits mentioned in the message. If the sum of the digits is greater than 9, check whether the sum forms a palindrome number.

Your task is to help Emily automate this process by writing a program that extracts all digits from a given message, calculates their sum, and checks if the sum is a palindrome if it is greater than 9.

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

The input consists of a string *s*, representing the customer message, which may

contain letters, digits, spaces, and other characters.

### **Output Format**

The output prints an integer representing the sum of all digits in the string, followed by a space.

If the sum is greater than 9, print "Palindrome" if the sum is a palindrome, otherwise print "Not palindrome".

If the sum is less than or equal to 9, no palindrome check is required.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 12 books 4 pen

Output: 7

### **Answer**

# You are using Python

```
a=input()
```

```
c=0
```

```
for i in a:
```

```
    if i.isdigit():
```

```
        c+=int(i)
```

```
print(c,end=' ')
```

```
if(c > 9):
```

```
    if str(c)==str(c)[::-1]:
```

```
        print("Palindrome")
```

```
    else:
```

```
        print("Not Palindrome")
```

**Status : Correct**

**Marks : 10/10**

## **2. Problem Statement**



Raja needs a program that helps him manage his shopping list efficiently. The program should allow him to perform the following operations:

**Add Items:** Raja should be able to add multiple items to his shopping list at once. He will input a space-separated list of items, each item being a string.

**Remove Item:** Raja should be able to remove a specific item from his shopping list. He will input the item he wants to remove, and if it exists in the list, it will be removed. If the item is not found, the program should notify him.

**Update List:** Raja might realize he forgot to add some items initially. After removing unnecessary items, he should be able to update his list by adding more items. Similar to the initial input, he will provide a space-separated list of new items.

### ***Input Format***

The first line consists of the initial list of integers should be entered as space-separated values.

The second line consists of the element to be removed should be entered as a single integer value.

The third line consists of the new elements to be appended should be entered as space-separated values.

### ***Output Format***

The output displays the current state of Raja's shopping list after each operation. After adding items, removing items, and updating the list, the program prints the updated shopping list in the following format:

"List1: [element1, element2, ... ,element\_n]"

List after removal: [element1, element2, ... ,element\_n]"

Final list: [element1, element2, ... ,element\_n]".

If the item is not found in the removing item process, print the message "Element not found in the list".

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 1 2 3 4 5

3

6 7 8

Output: List1: [1, 2, 3, 4, 5]

List after removal: [1, 2, 4, 5]

Final list: [1, 2, 4, 5, 6, 7, 8]

### **Answer**

```
a=list(map(int,input().split()))
```

```
b=int(input())
```

```
c=list(map(int,input().split()))
```

```
print("List1:",a)
```

```
if b in a:
```

```
    a.remove(b)
```

```
else:
```

```
    print("Element not found in the list")
```

```
print("List after removal:",a)
```

```
a.extend(c)
```

```
print("Final List:",a)
```

**Status :** Partially correct

**Marks :** 6.5/10

### **3. Problem Statement**

You have two strings str1 and str2, both of equal length.

Write a Python program to concatenate the two strings such that the first character of str1 is followed by the first character of str2, the second character of str1 is followed by the second character of str2, and so on.

For example, if str1 is "abc" and str2 is "def", the output should be "adbecf".

### ***Input Format***

The input consists of two strings in each line.

### ***Output Format***

The output displays the concatenated string in the mentioned format.

Refer to the sample output for formatting specifications.

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

Input: abc

def

Output: adbecf

### ***Answer***

```
# You are using Python
```

```
a=input()
```

```
b=input()
```

```
c=list()
```

```
n=len(a)
```

```
for i in range(n):
```

```
    c.append(a[i])
```

```
    c.append(b[i])
```

```
print("".join(c))
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 4\_MCQ

Attempt : 1  
Total Mark : 15  
Marks Obtained : 12

#### Section 1 : MCQ

1. What will be the output of the following Python code?

```
def cube(x):  
    return x * x * x  
x = cube(3)  
print(x)
```

**Answer**

27

**Status : Correct**

**Marks : 1/1**

2. What will be the output of the following Python code?

```
multiply = lambda x, y: x * y
```

```
print(multiply(2, 'Hello'))
```

**Answer**

HelloHello

**Status :** Correct

**Marks :** 1/1

3. What is the output of the code shown below?

```
def f1(x):  
    x += 1  
    print(x)
```

```
global_variable = 15  
f1(global_variable)  
print("hello")
```

**Answer**

16hello

**Status :** Correct

**Marks :** 1/1

4. What will be the output of the following code?

```
def display(*args):  
    for arg in args:  
        print(arg)
```

```
display(10, 20, 30)
```

**Answer**

102030

**Status :** Correct

**Marks :** 1/1

5. What is the output of the code shown?

```
def f():  
    global a
```

```
print(a)
a = "hello"
print(a)
a = "world"
f()
print(a)
```

**Answer**

worldhellohello

**Status :** Correct

**Marks :** 1/1

6. What will be the output of the following code?

```
value = 42
result = abs(value) + len(str(value))
print(result)
```

**Answer**

44

**Status :** Correct

**Marks :** 1/1

7. Which of the following functions can take a lambda function as a parameter in Python?

**Answer**

All of the mentioned options

**Status :** Wrong

**Marks :** 0/1

8. What is the output of the following code snippet?

```
def add(a, b=2):
    return a - b
```

```
result = add(3)
print(result)
```

**Answer**

1

**Status :** Correct

**Marks :** 1/1

9. What is the output of the following code snippet?

```
def my_function(x):  
    x += 5  
    return x
```

```
a = 10  
result = my_function(a)  
print(a, result)
```

**Answer**

10 15

**Status :** Correct

**Marks :** 1/1

10. What will be the output of the following Python code?

```
def func(a, b=5, c=10):  
    print('a is', a, 'and b is', b, 'and c is', c)  
  
func(3, 7)  
func(25, c = 24)  
func(c = 50, a = 100)
```

**Answer**

a is 3 and b is 7 and c is 10  
a is 25 and b is 5 and c is 24  
a is 100 and b is 5 and c is 50

**Status :** Correct

**Marks :** 1/1

11. What will be the output of the following Python code?

```
def is_even(number):
```

```
if number % 2 == 0:  
    return True
```

```
result = is_even(6)  
print(result)
```

**Answer**

True

**Status :** Correct

**Marks :** 1/1

12. What is the output of the following code?

```
x=12  
def f1(a,b=x):  
    print(a,b)  
x=15  
f1(4)
```

**Answer**

4 15

**Status :** Wrong

**Marks :** 0/1

13. What is the main advantage of using lambda functions in Python?

**Answer**

They allow you to write shorter code than regular functions

**Status :** Correct

**Marks :** 1/1

14. How is a lambda function different from a regular named function in Python?

**Answer**

A lambda function can only return one value, while a regular function can return multiple values

**Status :** Wrong

**Marks :** 0/1



15. What will be the output of the following Python code?

```
def display(b, n):  
    while n > 0:  
        print(b,end="")  
        n=n-1  
display('z',3)
```

**Answer**

zzz

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 4\_COD\_Updated

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

#### Section 1 : Coding

##### 1. Problem Statement

Sara is developing a text-processing tool that checks if a given string starts with a specific character or substring. She needs to implement a function that accepts a string and a character (or substring), and returns True if the string starts with the provided character/substring, or False otherwise.

Write a program that uses a lambda function to help Sara perform this check.

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

The first line contains a string `str` representing the main string to be checked.

The second line contains a string `n`, which is the character or substring to check if the main string starts with it.

### **Output Format**

The first line of output prints "True" if the string starts with the given character/substring, otherwise prints "False".

Refer to the sample for the formatting specifications.

### **Sample Test Case**

Input: Examly

e

Output: False

### **Answer**

```
a=input()
b=input()
p=lambda a,b:True if a.startswith(b) else False
r=p(a,b)
print(r)
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

Sneha is building a more advanced exponential calculator. She wants to implement a program that does the following:

Calculates the result of raising a given base to a specific exponent using Python's built-in pow() function. Displays all intermediate powers from base<sup>1</sup> to base<sup>exponent</sup> as a list. Calculates and displays the sum of these intermediate powers.

Help her build this program to automate her calculations.

### **Input Format**

The input consists of line-separated two integer values representing base and exponent.

### **Output Format**

The first line of the output prints the calculated result of raising the base to the exponent.

The second line prints a list of all powers from  $\text{base}^1$  to  $\text{base}^{\text{exponent}}$ .

The third line prints the sum of all these powers.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 2

3

Output: 8

[2, 4, 8]

14

### **Answer**

# You are using Python

```
a=int(input())
```

```
b=int(input())
```

```
print(pow(a,b))
```

```
s1=[]
```

```
for i in range(1,b+1):
```

```
    c=pow(a,i)
```

```
    s1.append(c)
```

```
print(s1)
```

```
print(sum(s1))
```

**Status :** Correct

**Marks :** 10/10

### **3. Problem Statement**

Imagine you are building a messaging application, and you want to know the length of the messages sent by the users. You need to create a program that calculates the length of a message using the built-in function `len()`.

### ***Input Format***

The input consists of a string representing the message.

### ***Output Format***

The output prints an integer representing the length of the entered message.

Refer to the sample output for formatting specifications.

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

Input: hello!!

Output: 7

### ***Answer***

```
# You are using Python
n=input()
print(len(n))
```

**Status :** Correct

**Marks :** 10/10

## **4. Problem Statement**

Implement a program that needs to identify Armstrong numbers.

Armstrong numbers are special numbers that are equal to the sum of their digits, each raised to the power of the number of digits in the number.

Write a function `is_armstrong_number(number)` that checks if a given number is an Armstrong number or not.

Function Signature: `armstrong_number(number)`

### ***Input Format***

The first line of the input consists of a single integer, `n`, representing the number to be checked.

### ***Output Format***

The output should consist of a single line that displays a message indicating whether the input number is an Armstrong number or not.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 153

Output: 153 is an Armstrong number.

### **Answer**

```
# You are using Python
n=int(input())
def armstrong_number(n):
    k=n
    a=len(str(n))
    count=0
    while n>0:
        rem=n%10
        count+=pow(rem,a)
        n//=10
    if count == k:
        print(k,"is an Armstrong number.")
    else:
        print(k,"is not an Armstrong number.")
armstrong_number(n)
```

**Status :** Correct

**Marks :** 10/10

## **5. Problem Statement**

Imagine you are developing a text analysis tool for a cybersecurity company. Your task is to create a function that analyzes input strings to categorize and count the characters into four categories: uppercase letters, lowercase letters, digits, and special characters. The company needs this tool to process log files and identify potential security threats.

Function Signature: `analyze_string(input_string)`

### ***Input Format***

The input consists of a single string (without space), which may include uppercase letters, lowercase letters, digits, and special characters.

### ***Output Format***

The first line contains an integer representing the count of uppercase letters in the format "Uppercase letters: [count]".

The second line contains an integer representing the count of lowercase letters in the format "Lowercase letters: [count]".

The third line contains an integer representing the count of digits in the format "Digits: [count]".

The fourth line contains an integer representing the count of special characters in the format "Special characters: [count]".

Refer to the sample output for the formatting specifications.

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

Input: Hello123

Output: Uppercase letters: 1

Lowercase letters: 4

Digits: 3

Special characters: 0

### ***Answer***

```
def analyze_string(input_string):
```

```
# You are using Python
```

```
#Type your code here
```

```
u=0
```

```
l=0
```

```
d=0
```

```
s=0
```

```
for i in (input_string):
```

```
    if i.isupper():
```

```
        u+=1
```

```
        elif i.islower():
            l+=1
        elif i.isdigit():
            d+=1
        else:
            s+=1
    return u,l,d,s

input_string = input()
uppercase_count, lowercase_count, digit_count, special_count =
analyze_string(input_string)

print("Uppercase letters:", uppercase_count)
print("Lowercase letters:", lowercase_count)
print("Digits:", digit_count)
print("Special characters:", special_count)
```

**Status :** Correct

**Marks :** 10/10



# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 4\_PAH\_Updated

Attempt : 1  
Total Mark : 60  
Marks Obtained : 60

#### Section 1 : Coding

##### 1. Problem Statement

Sophia is developing a feature for her online banking application that calculates the total sum of digits in customers' account numbers. This sum is used to generate unique verification codes for secure transactions. She needs a program that takes an account number as input and outputs the sum of its digits.

Help Sophia to complete her task.

Function Specification: `def sum_digits(num)`

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

The input consists of an integer, representing the customer's account number.

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

The output prints an integer representing the sum of the digits of the account number.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 123245

Output: 17

### **Answer**

```
num = int(input())
# You are using Python
def sum_digits(num):
    add=0
    temp=num
    while temp>0:
        digit=temp%10
        add+=digit
        temp//=10
    return add
```

```
sum = sum_digits(num)
print(sum)
```

**Status : Correct**

**Marks : 10/10**

## **2. Problem Statement**

Alice works at a digital marketing company, where she analyzes large datasets. One day, she's tasked with processing customer ID numbers, which are long numeric sequences.

To simplify her task, Alice needs to calculate the digital root of each ID. The digital root is obtained by repeatedly summing the digits of a number until a single digit remains.

Help Alice write a program that reads a customer ID number, calculates its digital root, and prints the result using a loop-based approach.

For example, the sum of the digits of 98675 is  $9 + 8 + 6 + 7 + 5 = 35$ , then  $3 + 5 = 8$ , which is the digital root.

Function prototype: `def digital_root(num)`

### ***Input Format***

The input consists of an integer num.

### ***Output Format***

The output prints an integer representing the sum of digits for a given number until a single digit is obtained.

Refer to the sample output for the formatting specifications.

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

Input: 451110

Output: 3

### ***Answer***

```
num = int(input())
# You are using Python
def digital_root(num):
    add=0
    temp=num
    while temp>0:
        digit=temp%10
        add+=digit
        temp//=10

    if(add>=10):
        add2=sum(int(i) for i in str(add))
```

```
        return add2
    else:
        print(add)
```

```
print(digital_root(num))
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

Ravi is working on analyzing a set of integers to determine how many of them are divisible by 3 and how many are divisible by 5. He decides to use lambda functions to filter and count the numbers based on their divisibility.

Write a program that takes a list of integers, calculates how many numbers are divisible by 3, and how many are divisible by 5, and then prints the results.

Additionally, the program should calculate the total sum of all numbers divisible by 3 and divisible by 5 separately.

#### **Input Format**

The first line contains an integer  $n$ , representing the number of integers in the list.

The second line contains  $n$  space-separated integers.

#### **Output Format**

The first line should print the count of numbers divisible by 3.

The second line should print the count of numbers divisible by 5.

The third line should print the sum of numbers divisible by 3.

The fourth line should print the sum of numbers divisible by 5.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 6

3 5 6 10 15 20

Output: 3

4

24

50

### **Answer**

# You are using Python

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

```
a=list(map(int,input().split()))
```

```
b= list(filter(lambda a:a%3==0,a))
```

```
c= list(filter(lambda a:a%5==0,a))
```

```
print(len(b))
```

```
print(len(c))
```

```
print(sum(b))
```

```
print(sum(c))
```

**Status :** Correct

**Marks :** 10/10

## **4. Problem Statement**

Create a Python program to monitor temperatures in a greenhouse using two sensors. Calculate and display the absolute temperature difference between the two sensor readings to ensure proper temperature control.

Note: Use the `abs()` built-in function.

### **Input Format**

The first line consists of a floating-point number, representing the temperature reading from Sensor 1.

The second line consists of a floating-point number, representing the temperature reading from Sensor 2.

### **Output Format**

The output displays the absolute temperature difference between Sensor 1 and

Sensor 2, rounded to two decimal places.

Refer to the sample output for the exact format.

**Sample Test Case**

Input: 33.2

26.7

Output: Temperature difference: 6.50 °C

**Answer**

```
# You are using Python
```

```
a=float(input())
```

```
b=float(input())
```

```
c=a-b
```

```
d=abs(c)
```

```
print(f"Temperature difference: {d:.2f} °C")
```

**Status :** Correct

**Marks :** 10/10

## 5. Problem Statement

Ella is designing a messaging application that needs to handle long text messages efficiently. To optimize storage and transmission, she plans to implement a text compression feature that replaces consecutive repeated characters with the character followed by its count, while leaving non-repeated characters unchanged.

Help Ella create a recursive function to achieve this compression without altering the original message's meaning.

Function Specification: `def compress_string(*args)`

**Input Format**

The input consists of a single line containing the string to be compressed.

**Output Format**

The output consists of a single line containing the compressed string.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: aaaBBBccc

Output: a3B3c3

### **Answer**

# You are using Python

```
def compress_string(args):
```

```
    s=""
```

```
    c=1
```

```
    args=args+''
```

```
    for i in range(len(args)-1):
```

```
        if(args[i]==args[i+1]):
```

```
            c+=1
```

```
        else:
```

```
            if(c==1):
```

```
                s+=args[i]
```

```
            else:
```

```
                s+=args[i]
```

```
                s+=str(c)
```

```
                c=1
```

```
    print(s)
```

```
b=input()
```

```
compress_string(b)
```

**Status :** Correct

**Marks :** 10/10

## **6. Problem Statement**

Hussain wants to create a program to calculate a person's BMI (Body Mass Index) based on their weight in kilograms and height in meters. The

BMI is a measure of a person's body fat relative to their height.

Your program should take user input for weight and height, calculate the BMI, and display the result.

Function Signature: `calculate_bmi(weight, height)`

Formula:  $BMI = \text{Weight} / (\text{Height})^2$

### ***Input Format***

The first line of input consists of a positive floating-point number, the person's weight in kilograms.

The second line of input consists of a positive floating-point number, the person's height in meters.

### ***Output Format***

The output displays "Your BMI is: [BM]" followed by a float value representing the calculated BMI, rounded off two decimal points.

Refer to the sample output for the formatting specifications.

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

Input: 70.0

1.75

Output: Your BMI is: 22.86

### ***Answer***

```
weight = float(input())
```

```
height = float(input())
```

```
# You are using Python
```

```
def calculate_bmi(weight,height):
```

```
    BMI= weight/(height)**2
```

```
    print(f"Your BMI is: {BMI:.2f}")
```

```
calculate_bmi(weight, height)
```



Status : Correct

Marks : 10/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 4\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

### Section 1 : Coding

#### 1. Problem Statement

Develop a text analysis tool that needs to count the occurrences of a specific substring within a given text string.

Write a function `count_substrings(text, substring)` that takes two inputs: the text string and the substring to be counted. The function should count how many times the substring appears in the text string and return the count.

Function Signature: `count_substrings(text, substring)`

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

The first line of the input consists of a string representing the text.

The second line consists of a string representing the substring.

### **Output Format**

The output should display a single line of output containing the count of occurrences of the substring in the text string.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: programming is fun and programming is cool  
programming

Output: The substring 'programming' appears 2 times in the text.

### **Answer**

```
# You are using Python
text=input()
substring=input()
def count_substrings(text,substring):
    n=len(substring)
    c=0
    for i in range(len(text)):
        if substring==text[i:n]:
            c+=1
            n+=1
    print("The substring '",substring,'" appears", c,"times in the text.");
count_substrings(text,substring)
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

Amrita is developing a password strength checker for her website. She wants the checker to consider the length and the diversity of characters

used in the password. A strong password should be long and include a mix of character types: uppercase, lowercase, digits, and special symbols.

She also wants the feedback to be user-friendly, so she wants to include the actual password in the output. Help Amrita finish this password checker using Python's built-in string methods.

Character Types Considered:

Lowercase letters (a-z) Uppercase letters (A-Z) Digits (0-9) Special characters (from string.punctuation, e.g. @, !, #, \$)

### ***Input Format***

The input consists of a single string representing the user's password.

### ***Output Format***

The program prints the strength of the password in this format:

If the password length < 6 characters or fewer than 2 of the 4 character types, the output prints "<password> is Weak"

If password length  $\geq 6$  and at least 2 different character types, the output prints "<password> is Moderate"

If Password length  $\geq 10$  and all 4 character types present, the output prints "<password> is Strong"

Refer to the sample output for formatting specifications.

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

Input: password123

Output: password123 is Moderate

### ***Answer***

# You are using Python

a=input()

u,l,d,s=0,0,0,0

for i in a:

```
if i.isupper():
    u+=1
elif i.islower():
    l+=1
elif i.isdigit():
    d+=1
else:
    s+=1
ch=sum([u>0,l>0,d>0,s>0])
if len(a)>=10 and ch==4 :
    print(a,"is Strong")
elif len(a)>=6 and ch>=2:
    print(a,"is Moderate")
else:
    print(a,"is Weak")
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

Imagine you are tasked with developing a function for calculating the total cost of an item after applying a sales tax. The sales tax rate is equal to 0.08 and it is defined as a global variable.

The function should accept the cost of the item as a parameter, calculate the tax amount, and return the total cost.

Additionally, the program should display the item cost, sales tax rate, and total cost to the user.

Function Signature: total\_cost(item\_cost)

#### **Input Format**

The input consists of a single line containing a positive floating-point number representing the cost of the item.

#### **Output Format**

The output consists of three lines:

"Item Cost:" followed by the cost of the item formatted to two decimal places.

"Sales Tax Rate:" followed by the sales tax rate in percentage.

"Total Cost:" followed by the calculated total cost after applying the sales tax, formatted to two decimal places.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 50.00

Output: Item Cost: \$50.00

Sales Tax Rate: 8.0%

Total Cost: \$54.00

### **Answer**

```
#
```

```
# You are using Python
```

```
item_cost=float(input())
```

```
SALES_TAX_RATE=0.08
```

```
def total_cost(item_cost):
```

```
    t=item_cost+(SALES_TAX_RATE*item_cost)
```

```
    return t
```

```
total_cost = total_cost(item_cost)
```

```
print(f"Item Cost: ${item_cost:.2f}")
```

```
print(f"Sales Tax Rate: {SALES_TAX_RATE * 100}%")
```

```
print(f"Total Cost: ${total_cost:.2f}")
```

**Status :** Correct

**Marks : 10/10**

## **4. Problem Statement**

Meena is analyzing a list of integers and needs to count how many numbers in the list are even and how many are odd. She decides to use

lambda functions to filter the even and odd numbers from the list.

Write a program that takes a list of integers, counts the number of even and odd numbers using lambda functions, and prints the results.

***Input Format***

The first line contains an integer n, representing the number of integers in the list.

The second line contains n space-separated integers.

***Output Format***

The first line of output prints an integer representing the count of even numbers.

The second line of output prints an integer representing the count of odd numbers.

Refer to the sample output for the formatting specifications.

***Sample Test Case***

Input: 7  
12 34 56 78 98 65 23

Output: 5  
2

***Answer***

```
# You are using Python
n=int(input())
a=list(map(int,input().split()))
s= len(list(filter(lambda a:a%2==0,a)))
r= len(list(filter(lambda a:a%2!=0,a)))
print(s)
print(r)
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_MCQ

Attempt : 1  
Total Mark : 20  
Marks Obtained : 15

#### Section 1 : MCQ

1. Fill in the code in order to get the following output.

Output:

Tuple: (1, 3, 4)

Max value: 4

t=(1,)

```
_____  
print("Tuple:" ,t)  
print("Max value:",_____)
```

**Answer**

1) t=t+(3,4)2) t.max()

**Status : Wrong**

**Marks : 0/1**



2. What will be the output?

```
a={'B':5,'A':9,'C':7}
print(sorted(a))
```

**Answer**

['A', 'B', 'C'].

**Status :** Correct

**Marks :** 1/1

3. Set  $s1 = \{1, 2, 4, 3\}$  and  $s2 = \{1, 5, 4, 6\}$ , find  $s1 \& s2$ ,  $s1 - s2$ ,  $s1 \mid s2$  and  $s1 \wedge s2$ .

**Answer**

$s1 \& s2 = \{1, 4\}$   $s1 - s2 = \{2, 3\}$   $s1 \wedge s2 = \{2, 3, 5, 6\}$   $s1 \mid s2 = \{1, 2, 3, 4, 5, 6\}$

**Status :** Correct

**Marks :** 1/1

4. Which of the following statements is used to create an empty tuple?

**Answer**

()

**Status :** Correct

**Marks :** 1/1

5. Predict the output of the following Python program

```
init_tuple_a = 1, 2, 8
init_tuple_b = (1, 2, 7)
set1=set(init_tuple_b)
set2=set(init_tuple_a)
print (set1 | set2)
print (init_tuple_a | init_tuple_b)
```

**Answer**

{1, 2, 7, 8}TypeError: unsupported operand type

**Status :** Correct

**Marks :** 1/1

6. What is the output of the following?

```
set1 = {10, 20, 30, 40, 50}
set2 = {60, 70, 10, 30, 40, 80, 20, 50}
print(set1.issubset(set2))
print(set2.issuperset(set1))
```

**Answer**

TrueTrue

**Status :** Correct

**Marks :** 1/1

7. What is the output of the following code?

```
a={"a":1,"b":2,"c":3}
b=dict(zip(a.values(),a.keys()))
print(b)
```

**Answer**

{1: 'a', 2: 'b', 3: 'c'}

**Status :** Correct

**Marks :** 1/1

8. What will be the output for the following code?

```
a=(1,2,3)
b=('A','B','C')
c=zip(a,b)
```

```
print(c)
print(tuple(c))
```

**Answer**

((1, 'A'), (2, 'B'), (3, 'C'))

**Status :** Correct

**Marks :** 1/1

9. Which of the statements about dictionary values is false?

**Answer**

Values of a dictionary must be unique

**Status :** Correct

**Marks :** 1/1

10. What is the result of `print(type({}))` is set)?

**Answer**

False

**Status :** Correct

**Marks :** 1/1

11. What is the output of the following code?

```
a={1:"A",2:"B",3:"C"}  
b=a.copy()  
b[2]="D"  
print(a)
```

**Answer**

{1: 'A', 2: 'D', 3: 'C'}

**Status :** Wrong

**Marks :** 0/1

12. Which of the following isn't true about dictionary keys?

**Answer**

Keys must be integers

**Status :** Correct

**Marks :** 1/1

13. What will be the output for the following code?

```
t1 = (1, 2, 4, 3)  
t2 = (1, 2, 3, 4)  
print(t1 < t2)
```

**Answer**

False

**Status : Correct**

**Marks : 1/1**

14. What will be the output of the following code?

```
a=(1,2,3,4)
print(sum(a,3))
```

**Answer**

Too many arguments for sum() method

**Status : Wrong**

**Marks : 0/1**

15. What is the output of the below Python code?

```
list1 = [1, 2, 3]
list2 = [5, 6, 7]
list3 = [10, 11, 12]
set1 = set(list2)
set2 = set(list1)
set1.update(set2)
set1.update(list3)
print(set1)
```

**Answer**

{1, 2, 3, 5, 6, 7, 10, 11, 12}

**Status : Correct**

**Marks : 1/1**

16. Which of the following is a Python tuple?

**Answer**

(1, 2, 3)

**Status : Correct**

**Marks : 1/1**

17. If 'a' is a dictionary with some key-value pairs, what does a.popitem() do?

**Answer**

Removes the key-value pair for the key given as an argument

**Status : Wrong**

**Marks : 0/1**

18. What will be the output of the following program?

```
set1 = {1, 2, 3}
set2 = set1.copy()
set2.add(4)
print(set1)
```

**Answer**

{1, 2, 3}

**Status : Correct**

**Marks : 1/1**

19. Suppose t = (1, 2, 4, 3), which of the following is incorrect?

**Answer**

t[3] = 45

**Status : Correct**

**Marks : 1/1**

20. What is the output of the following code?

```
a=(1,2,(4,5))
b=(1,2,(3,4))
print(a<b)
```

**Answer**

Error, &lt; operator is not valid for tuples

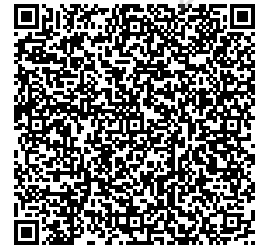
**Status : Wrong**

**Marks : 0/1**

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_COD

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

### Section 1 : Coding

#### 1. Problem Statement

Ella is analyzing the sales data for a new online shopping platform. She has a record of customer transactions where each customer's data includes their ID and a list of amounts spent on different items. Ella needs to determine the total amount spent by each customer and identify the highest single expenditure for each customer.

Your task is to write a program that computes these details and displays them in a dictionary.

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

The first line of input consists of an integer  $n$ , representing the number of customers.

Each of the next n lines contains a numerical customer ID followed by integers representing the amounts spent on different items.

### **Output Format**

The output displays a dictionary where the keys are customer IDs and the values are lists containing two integers: the total expenditure and the maximum single expenditure.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 2

101 100 150 200

102 50 75 100

Output: {101: [450, 200], 102: [225, 100]}

### **Answer**

# You are using Python

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

```
cd={}
```

```
for i in range(n):
```

```
    d=list(map(int,input().split()))
```

```
    c=d[0]
```

```
    a=d[1:]
```

```
    t=sum(a)
```

```
    m=max(a)
```

```
    cd[c]=[t,m]
```

```
print(cd)
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

Professor Adams needs to analyze student participation in three recent academic workshops. She has three sets of student IDs: the first set contains students who registered for the workshops, the second set

contains students who actually attended, and the third set contains students who dropped out.

Professor Adams needs to determine which students who registered also attended, and then identify which of these students did not drop out.

Help Professor Adams identify the students who registered, attended, and did not drop out of the workshops.

### ***Input Format***

The first line of input consists of integers, representing the student IDs who registered for the workshops.

The second line consists of integers, representing the student IDs who attended the workshops.

The third line consists of integers, representing the student IDs who dropped out of the workshops.

### ***Output Format***

The first line of output displays the intersection of the first two sets, which shows the IDs of students who registered and attended.

The second line displays the result after removing student IDs that are in the third set (dropped out), showing the IDs of students who both attended and did not drop out.

Refer to the sample output for the formatting specifications.

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

Input: 1 2 3

2 3 4

3 4 5

Output: {2, 3}

{2}

### ***Answer***



```
# You are using Python
r=set(map(int,input().split()))
a=set(map(int,input().split()))
d=set(map(int,input().split()))
```

```
a1=r&a
a2=a1-d
```

```
print(a1)
print(a2)
```

**Status :** Correct

**Marks : 10/10**

### 3. Problem Statement

James is managing a list of inventory items in a warehouse. Each item is recorded as a tuple, where the first element is the item ID and the second element is a list of quantities available for that item. James needs to filter out all quantities that are above a certain threshold to find items that have a stock level above this limit.

Help James by writing a program to process these tuples, filter the quantities from all the available items, and display the results.

Note:

Use the filter() function to filter out the quantities greater than the specified threshold for each item's stock list.

#### **Input Format**

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

The next N lines each contain a tuple in the format (ID, [quantity1, quantity2, ...]), where ID is an integer and the list contains integers.

The final line consists of an integer threshold, representing the quantity threshold.

#### **Output Format**

The output should be a single line displaying the filtered quantities, space-

separated. Each quantity is strictly greater than the given threshold.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 2

(1, [1, 2])

(2, [3, 4])

2

Output: 3 4

### **Answer**

# You are using Python

```
N=int(input())
```

```
a=[]
```

```
for i in range(N):
```

```
    it=eval(input())
```

```
    q=it[1]
```

```
    a.extend(q)
```

```
t=int(input())
```

```
f=list(filter(lambda x: x>t,a))
```

```
for j in f:
```

```
    print(j,end=' ')
```

**Status :** Correct

**Marks :** 10/10

## **4. Problem Statement**

Liam is analyzing a list of product IDs from a recent sales report. He needs to determine how frequently each product ID appears and calculate the following metrics:

Frequency of each product ID: A dictionary where the key is the product ID and the value is the number of times it appears. Total number of unique product IDs. Average frequency of product IDs: The average count of all product IDs.

Write a program to read the product IDs, compute these metrics, and output the results.

Example

Input:

6 //number of product ID

101

102

101

103

101

102 //product IDs

Output:

{101: 3, 102: 2, 103: 1}

Total Unique IDs: 3

Average Frequency: 2.00

Explanation:

Input 6 indicates that you will enter 6 product IDs.

A dictionary is created to track the frequency of each product ID.

Input 101: Added with a frequency of 1.

Input 102: Added with a frequency of 1.

Input 101: Frequency of 101 increased to 2.

Input 103: Added with a frequency of 1.

Input 101: Frequency of 101 increased to 3.

Input 102: Frequency of 102 increased to 2.

The dictionary now contains 3 unique IDs: 101, 102, and 103.

Total Unique is 3.

The average frequency is 2.00.

### ***Input Format***

The first line of input consists of an integer  $n$ , representing the number of product IDs.

The next  $n$  lines each contain a single integer, each representing a product ID.

### ***Output Format***

The first line of output displays the frequency dictionary, which maps each product ID to its count.

The second line displays the total number of unique product IDs, preceded by "Total Unique IDs: ".

The third line displays the average frequency of the product IDs. This is calculated by dividing the total number of occurrences of all product IDs by the total number of unique product IDs, rounded to two decimal places. It is preceded by "Average Frequency: ".

Refer to the sample output for formatting specifications.

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

Input: 6

101

102

101

103

101

102

Output: {101: 3, 102: 2, 103: 1}

Total Unique IDs: 3

Average Frequency: 2.00

### ***Answer***

# You are using Python

```

n=int(input())
f={}
for i in range(n):
    p=int(input())
    if p in f:
        f[p]+=1
    else:
        f[p]=1
u=len(f)
t=sum(f.values())
a=t/u
print(f)
print("Total Unique IDs:",u)
print(f"Average Frequency:{a:.2f}")

```

**Status :** Correct

**Marks :** 10/10

## 5. Problem Statement

Gowshik is working on a task that involves taking two lists of integers as input, finding the element-wise sum of the corresponding elements, and then creating a tuple containing the sum values.

Write a program to help Gowshik with this task.

Example:

Given list:

[1, 2, 3, 4]

[3, 5, 2, 1]

An element-wise sum of the said tuples: (4, 7, 5, 5)

### **Input Format**

The first line of input consists of a single integer  $n$ , representing the length of the input lists.

The second line of input consists of  $n$  integers separated by commas,

representing the elements of the first list.

The third line of input consists of n integers separated by commas, representing the elements of the second list.

### ***Output Format***

The output is a single line containing a tuple of integers separated by commas, representing the element-wise sum of the corresponding elements from the two input lists.

Refer to the sample output for the formatting specifications.

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

Input: 4

1, 2, 3, 4

3, 5, 2, 1

Output: (4, 7, 5, 5)

### ***Answer***

# You are using Python

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

```
a=list(map(int,input().split(',')))
```

```
b=list(map(int,input().split(',')))
```

```
t=tuple(map(sum,zip(a,b)))
```

```
print(t)
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_PAH

Attempt : 1  
Total Mark : 60  
Marks Obtained : 46

### Section 1 : Coding

#### 1. Problem Statement

Tom wants to create a dictionary that lists the first n prime numbers, where each key represents the position of the prime number, and the value is the prime number itself.

Help Tom generate this dictionary based on the input she provides.

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

The input consists of an integer n, representing the number of prime numbers Tom wants to generate.

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

The output displays the generated dictionary where each key is an integer from 1 to n, and the corresponding value is the prime number.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 4

Output: {1: 2, 2: 3, 3: 5, 4: 7}

### **Answer**

# You are using Python

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

```
def p(n):
```

```
    if n<=1:
```

```
        return False
```

```
    for i in range(2,int(n**0.5)+1):
```

```
        if n%i==0:
```

```
            return False
```

```
    return True
```

```
def gp(n):
```

```
    pr={}
```

```
    c=2
```

```
    co=1
```

```
    while len(pr) < n:
```

```
        if p(c):
```

```
            pr[co]=c
```

```
            co+=1
```

```
            c+=1
```

```
    return pr
```

```
print(gp(n))
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

Mia is organizing a list of integers into a series of pairs for his new project. She wants to create pairs of consecutive integers from the list. The last integer should be paired with None to complete the series. The pairing happens as follows: ((Element 1, Element 2), (Element 2, Element 3)).....



(Element n, None)).

Your task is to help Henry by writing a Python program that reads a list of integers, forms these pairs, and displays the result in tuple format.

### ***Input Format***

The first line of input consists of an integer n, representing the number of elements in the tuple.

The second line of input contains n space-separated integers, representing the elements of the tuple.

### ***Output Format***

The output displays a tuple containing pairs of consecutive integers from the input. The last integer in the tuple is paired with 'None'.

Refer to the sample output for formatting specifications.

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

Input: 3

5 10 15

Output: ((5, 10), (10, 15), (15, None))

### ***Answer***

```
# You are using Python
n=int(input())
e=list(map(int,input().split()))
r=tuple((e[i],e[i+1])for i in range(n-1))
r+=(e[-1],None,)
print(r)
```

**Status :** Correct

**Marks :** 10/10

## **3. Problem Statement**

Maya wants to create a dictionary that maps each integer from 1 to a given

number  $n$  to its square. She will use this dictionary to quickly reference the square of any number up to  $n$ .

Help Maya generate this dictionary based on the input she provides.

### ***Input Format***

The input consists of an integer  $n$ , representing the highest number for which Maya wants to calculate the square.

### ***Output Format***

The output displays the generated dictionary where each key is an integer from 1 to  $n$ , and the corresponding value is its square.

Refer to the sample output for formatting specifications.

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

Input: 5

Output: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

### ***Answer***

# You are using Python

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

```
s={i:i**2 for i in range(1,n+1)}
```

```
print(s)
```

**Status :** Correct

**Marks :** 10/10

## **4. Problem Statement**

Jordan is creating a program to process a list of integers. The program should take a list of integers as input, remove any duplicate integers while preserving their original order, concatenate the remaining unique integers into a single string, and then print the result.

Help Jordan in implementing the same.

### **Input Format**

The input consists of space-separated integers representing the elements of the set.

### **Output Format**

The output prints a single integer formed by concatenating the unique integers from the input in the order they appeared.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 11 11 33 50

Output: 113350

### **Answer**

```
# You are using Python
n=input().split()
se=set()
u=[]
for i in n:
    if i not in se:
        u.append(i)
        se.add(i)
print("".join(u))
```

**Status : Correct**

**Marks : 10/10**

## **5. Problem Statement**

Sophia is organizing a list of event IDs representing consecutive days of an event. She needs to group these IDs into consecutive sequences. For example, if the IDs 3, 4, and 5 appear consecutively, they should be grouped.

Write a program that helps Sophia by reading the total number of event IDs and the IDs themselves, then display each group of consecutive IDs in

tuple format.

### ***Input Format***

The first line of input consists of an integer n, representing the number of event IDs.

The next n lines contain integers representing the event IDs, where each integer corresponds to an event ID.

### ***Output Format***

The output should display each group of consecutive event IDs in a tuple format. Each group should be printed on a new line, and single event IDs should be displayed as a single-element tuple.

Refer to the sample output for formatting specifications.

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

Input: 3

1  
2  
3

Output: (1, 2, 3)

### ***Answer***

# You are using Python

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

```
a=[]
```

```
for i in range(n):
```

```
    num=int(input())
```

```
    a.append(num)
```

```
a.sort
```

```
group=[]
```

```
temp=[a[0]]
```

```
for i in range(1,n):
```

```
    if a[i]==a[i-1]+1:
```

```
        temp.append(a[i])
```

```
    else:
```

```
group.append(tuple(temp))
temp=[a[i]]
group.append(tuple(temp))
```

```
for g in group:
    print(g)
```

**Status :** Partially correct

**Marks :** 6/10

## 6. Problem Statement

Rishi is working on a program to manipulate a set of integers. The program should allow users to perform the following operations:

Find the maximum value in the set. Find the minimum value in the set. Remove a specific number from the set.

The program should handle these operations based on user input. If the user inputs an invalid operation choice, the program should indicate that the choice is invalid.

### **Input Format**

The first line contains space-separated integers that will form the initial set. Each integer  $x$  is separated by a space.

The second line contains an integer  $ch$ , representing the user's choice:

- 1 to find the maximum value
- 2 to find the minimum value
- 3 to remove a specific number from the set

If  $ch$  is 3, the third line contains an integer  $n1$ , which is the number to be removed from the set.

### **Output Format**

The first line of output prints the original set in descending order.

For choice 1: Print the maximum value from the set.

For choice 2: Print the minimum value from the set.

For choice 3: Print the set after removing the specified number, in descending order.

For invalid choices: Print "Invalid choice".

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: 1 2 3 4 5

1

Output: {5, 4, 3, 2, 1}

5

**Answer**

-

**Status :** Skipped

**Marks :** 0/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 5\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 10

### Section 1 : Coding

#### 1. Problem Statement

James is an engineer working on designing a new rocket propulsion system. He needs to solve a quadratic equation to determine the optimal launch trajectory. The equation is of the form  $ax^2 + bx + c = 0$ .

Your task is to help James find the roots of this quadratic equation. Depending on the discriminant, the roots might be real and distinct, real and equal, or complex. Implement a program to determine and display the roots of the equation based on the given coefficients.

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

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

The second line contains three space-separated integers a,b, and c representing the coefficients of the quadratic equation.

### **Output Format**

The output displays:

1. If the discriminant is positive, display the two real roots.
2. If the discriminant is zero, display the repeated real root.
3. If the discriminant is negative, display the complex roots as a tuple with real and imaginary parts.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 3

1 5 6

Output: (-2.0, -3.0)

### **Answer**

-

**Status :** Skipped

**Marks :** 0/10

## **2. Problem Statement**

Alex is working with grayscale pixel intensities from an old photo that has been scanned in a single row. To detect edges in the image, Alex needs to calculate the differences between each pair of consecutive pixel intensities.

Your task is to write a program that performs this calculation and returns the result as a tuple of differences.

### **Input Format**

The first line of input contains an integer n, representing the number of pixel intensities.



The second line contains n space-separated integers representing the pixel intensities.

### **Output Format**

The output displays a tuple containing the absolute differences between consecutive pixel intensities.

Refer to the sample output for format specifications.

### **Sample Test Case**

Input: 5

200 100 20 80 10

Output: (100, 80, 60, 70)

### **Answer**

```
# You are using Python
n=int(input())
pixel=list(map(int,input().split()))
diff=tuple(abs(pixel[i+1]-pixel[i]) for i in range (n-1))
print(diff)
```

**Status :** Correct

**Marks :** 10/10

## **3. Problem Statement**

Riya owns a store and keeps track of item prices from two different suppliers using two separate dictionaries. He wants to compare these prices to identify any differences. Your task is to write a program that calculates the absolute difference in prices for items that are present in both dictionaries. For items that are unique to one dictionary (i.e., not present in the other), include them in the output dictionary with their original prices.

Help Riya to implement the above task using a dictionary.

### **Input Format**

The first line of input consists of an integer  $n_1$ , representing the number of items in the first dictionary.

The next  $n_1$  lines contain two integers

1. The first line contains the item (key), and
2. The second line contains the price (value).

The following line consists of an integer  $n_2$ , representing the number of items in the second dictionary

The next  $n_2$  lines contain two integers

1. The first line contains the item (key), and
2. The second line contains the price (value).

### **Output Format**

The output should display a dictionary that includes:

1. For items common to both dictionaries, the absolute difference between their prices.
2. For items that are unique to one dictionary, the original price from that dictionary.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 1

4

4

1

8

7

Output: {4: 4, 8: 7}

### **Answer**

-

**Status :** Skipped

**Marks :** 0/10

#### 4. Problem Statement

Emily is a librarian who keeps track of books borrowed and returned by her patrons. She maintains four sets of book IDs: the first set represents books borrowed, the second set represents books returned, the third set represents books added to the collection, and the fourth set represents books that are now missing. Emily wants to determine which books are still borrowed but not returned, as well as those that were added but are now missing. Finally, she needs to find all unique book IDs from both results.

Help Emily by writing a program that performs the following operations on four sets of integers:

Compute the difference between the borrowed books (first set) and the returned books (second set). Compute the difference between the added books (third set) and the missing books (fourth set). Find the union of the results from the previous two steps, and sort the final result in descending order.

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

The first line of input consists of a list of integers representing borrowed books.

The second line of input consists of a list of integers representing returned books.

The third line of input consists of a list of integers representing added books.

The fourth line of input consists of a list of integers representing missing books.

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

The first line of output displays the difference between sets P and Q, sorted in descending order.

The second line of output displays the difference between sets R and S, sorted in descending order.

The third line of output displays the union of the differences from the previous two steps, sorted in descending order.

Refer to the sample output for the formatting specifications.

**Sample Test Case**

Input: 1 2 3

2 3 4

5 6 7

6 7 8

Output: [1]

[5]

[5, 1]

**Answer**

-

**Status :** Skipped

**Marks :** 0/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 6\_MCQ

Attempt : 1  
Total Mark : 20  
Marks Obtained : 18

#### Section 1 : MCQ

1. What will be the output of the following Python code?

```
# Predefined lines to simulate the file content
```

```
lines = [  
    "This is 1st line",  
    "This is 2nd line",  
    "This is 3rd line",  
    "This is 4th line",  
    "This is 5th line"  
]
```

```
print("Name of the file: foo.txt")
```

```
# Print the first 5 lines from the predefined list  
for index in range(5):  
    line = lines[index]
```

```
print("Line No %d - %s" % (index + 1, line.strip()))
```

**Answer**

Displays Output

**Status :** Correct

**Marks :** 1/1

2. What happens if no arguments are passed to the seek function?

**Answer**

error

**Status :** Wrong

**Marks :** 0/1

3. What is the default value of reference\_point in the following code?

```
file_object.seek(offset [,reference_point])
```

**Answer**

0

**Status :** Correct

**Marks :** 1/1

4. What is the correct way to raise an exception in Python?

**Answer**

raise Exception()

**Status :** Correct

**Marks :** 1/1

5. What is the output of the following code?

```
class MyError(Exception):  
    pass
```

```
try:
```

```
    raise MyError("Something went wrong")
```

```
except MyError as e:  
    print(e)
```

**Answer**

Something went wrong

**Status :** Correct

**Marks :** 1/1

6. What is the purpose of the except clause in Python?

**Answer**

To handle exceptions during code execution

**Status :** Correct

**Marks :** 1/1

7. Fill in the blanks in the following code of writing data in binary files.

```
import _____ (1)  
rec=[]  
while True:  
    rn=int(input("Enter"))  
    nm=input("Enter")  
    temp=[rn, nm]  
    rec.append(temp)  
    ch=input("Enter choice (y/N)")  
    if ch.upper=="N":  
        break  
f.open("stud.dat", "_____")(2)  
_____.dump(rec,f)(3)  
_____.close()(4)
```

**Answer**

(pickle,wb,pickle,f)

**Status :** Correct

**Marks :** 1/1

8. Match the following:

- a) f.seek(5,1) i) Move file pointer five characters behind from the current position
- b) f.seek(-5,1) ii) Move file pointer to the end of a file
- c) f.seek(0,2) iii) Move file pointer five characters ahead from the current position
- d) f.seek(0) iv) Move file pointer to the beginning of a file

**Answer**

a-iii, b-i, c-ii, d-iv

**Status : Correct**

**Marks : 1/1**

9. Which of the following is true about the finally block in Python?

**Answer**

The finally block is always executed, regardless of whether an exception occurs or not

**Status : Correct**

**Marks : 1/1**

10. How do you create a user-defined exception in Python?

**Answer**

By creating a new class that inherits from the Exception class

**Status : Correct**

**Marks : 1/1**

11. Fill in the code in order to get the following output:

Output:

Name of the file: ex.txt

```
fo = open(_____(1), "wb")  
print("Name of the file: ", _____)(2)
```

**Answer**



1) "ex.txt"2) fo.name

**Status :** Correct

**Marks :** 1/1

12. What is the difference between r+ and w+ modes?

**Answer**

in r+ the pointer is initially placed at the beginning of the file and the pointer is at the end for w+

**Status :** Correct

**Marks :** 1/1

13. How do you rename a file?

**Answer**

os.rename(existing\_name, new\_name)

**Status :** Correct

**Marks :** 1/1

14. Which of the following is true about  
fp.seek(10,1)

**Answer**

Move file pointer ten characters ahead from the current position

**Status :** Correct

**Marks :** 1/1

15. What happens if an exception is not caught in the except clause?

**Answer**

The program will display a traceback error and stop execution

**Status :** Correct

**Marks :** 1/1

16. Fill the code to in order to read file from the current position.

Assuming exp.txt file has following 3 lines, consider current file position is

beginning of 2nd line

Meri,25

John,21

Raj,20

Ouput:

['John,21\n','Raj,20\n']

```
f = open("exp.txt", "w+")
```

```
_____ (1)
```

```
print _____ (2)
```

**Answer**

1) f.seek(0, 1) 2) f.readlines()

**Status :** Correct

**Marks :** 1/1

17. What will be the output of the following Python code?

```
f = None
```

```
for i in range (5):
```

```
    with open("data.txt", "w") as f:
```

```
        if i > 2:
```

```
            break
```

```
print(f.closed)
```

**Answer**

True

**Status :** Correct

**Marks :** 1/1

18. Which clause is used to clean up resources, such as closing files in Python?

**Answer**

finally

**Status :** Correct

**Marks :** 1/1

19. What is the output of the following code?

```
try:  
    x = 1 / 0  
except ZeroDivisionError:  
    print("Caught division by zero error")  
finally:  
    print("Executed")
```

**Answer**

Caught division by zero errorExecuted

**Status :** Correct

**Marks :** 1/1

20. What is the output of the following code?

```
try:  
    x = "hello" + 5  
except TypeError:  
    print("Type Error occurred")  
finally:  
    print("This will always execute")
```

**Answer**

Type Error occurred

**Status :** Wrong

**Marks :** 0/1

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 6\_COD

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

### Section 1 : Coding

#### 1. Problem Statement

A retail store requires a program to calculate the total cost of purchasing a product based on its price and quantity. The program performs validation to ensure valid inputs and handles specific error conditions using exceptions:

Price Validation: If the price is zero or less, raise a ValueError with the message: "Invalid Price". Quantity Validation: If the quantity is zero or less, raise a ValueError with the message: "Invalid Quantity". Cost Threshold: If the total cost exceeds 1000, raise RuntimeError with the message: "Excessive Cost".

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

The first line of input consists of a double value, representing the price of a product.

The second line consists of an integer, representing the quantity of the product.

### **Output Format**

If the calculation is successful, print the total cost rounded to one decimal place.

If the price is zero or less prints "Invalid Price".

If the quantity is zero or less prints "Invalid Quantity".

If the total cost exceeds 1000, prints "Excessive Cost".

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 20.0

5

Output: 100.0

### **Answer**

# You are using Python

```
n=float(input())
```

```
a=int(input())
```

```
d=n*a
```

```
try:
```

```
    if n<=0:
```

```
        raise ValueError("Invalid Price")
```

```
    if a<=0:
```

```
        raise ValueError("Invalid Quantity")
```

```
    if d>1000:
```

```
        raise RuntimeError("Excessive Cost")
```

```
    print(d)
```

```
except ValueError as V:
```

```
    print(V)
```

```
except RuntimeError as r:
```

```
    print(r)
```

**Status : Correct**

**Marks : 10/10**

## 2. Problem Statement

Write a program that calculates the average of a list of integers. The program prompts the user to enter the length of the list (n) and each element of the list. It performs error handling to ensure that the length of the list is a non-negative integer and that each input element is a numeric value.

### ***Input Format***

The first line of the input is an integer n, representing the length of the list as a positive integer.

The second line of the input consists of an element of the list as an integer, separated by a new line.

### ***Output Format***

If the length of the list is not a positive integer or zero, the output displays "Error: The length of the list must be a non-negative integer."

If a non-numeric value is entered for the length of the list, the output displays "Error: You must enter a numeric value."

If a non-numeric value is entered for a list element, the output displays "Error: You must enter a numeric value."

If the inputs are valid, the program calculates and prints the average of the provided list of integers with two decimal places: "The average is: [average]".

Refer to the sample output for the formatting specifications.

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

Input: -2

1

2

Output: Error: The length of the list must be a non-negative integer.

### Answer

# You are using Python

try:

n\_input = input()

n = int(n\_input)

if n < 0 or n > 20:

print("Error: The length of the list must be a non-negative integer.")

else:

numbers = []

for i in range(n):

try:

element = input()

number = int(element)

numbers.append(number)

except ValueError:

print("Error: You must enter a numeric value.")

exit()

if n == 0:

print("The average is: 0.00")

else:

average = sum(numbers) / n

print(f"The average is: {average:.2f}")

except ValueError:

print("Error: You must enter a numeric value.")

**Status : Correct**

**Marks : 10/10**

### 3. Problem Statement

Tara is a content manager who needs to perform case conversions for various pieces of text and save the results in a structured manner.

She requires a program to take a user's input string, save it in a file, and then retrieve and display the string in both upper-case and lower-case versions. Help her achieve this task efficiently.

File Name: text\_file.txt

### ***Input Format***

The input consists of a single line containing a string provided by the user.

### ***Output Format***

The first line displays the original string read from the file in the format: "Original String: {original\_string}".

The second line displays the upper-case version of the original string in the format: "Upper-Case String: {upper\_case\_string}".

The third line displays the lower-case version of the original string in the format: "Lower-Case String: {lower\_case\_string}".

Refer to the sample output for the formatting specifications.

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

Input: #SpecialSymBoLs1234

Output: Original String: #SpecialSymBoLs1234

Upper-Case String: #SPECIALSYMBOLS1234

Lower-Case String: #specialsymbols1234

### ***Answer***

# You are using Python

```
a=input()
```

```
with open ("text.txt","w") as f:
```

```
    f.write(a)
```

```
with open ("text.txt","r") as f:
```

```
    c=f.read()
```

```
print(f"Original String: {c}")
```

```
d=c.upper()
```

```
e=c.lower()
```

```
print(f"Upper-Case String: {d}")
```

```
print(f"Lower-Case String: {e}")
```

**Status :** Correct

**Marks :** 10/10

## **4. Problem Statement**



In a voting system, a person must be at least 18 years old to be eligible to vote. If a user enters an age below 18, the system should raise a user-defined exception indicating that they are not eligible to vote.

### ***Input Format***

The input contains a positive integer representing age.

### ***Output Format***

If the age is less than 18, the output displays "Not eligible to vote".

Otherwise, the output displays "Eligible to vote".

Refer to the sample output for formatting specifications.

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

Input: 18

Output: Eligible to vote

### ***Answer***

# You are using Python

```
class NotEligibleToVote(Exception):  
    pass
```

```
try:  
    age_input = input()  
    age = int(age_input)  
  
    if age < 18:  
        raise NotEligibleToVote  
    else:  
        print("Eligible to vote")
```

```
except NotEligibleToVote:  
    print("Not eligible to vote")  
except ValueError:  
    print("Invalid input. Please enter a numeric value.")
```

Status : Correct

Marks : 10/10

## 5. Problem Statement

Sophie enjoys playing with words and wants to count the number of words in a sentence. She inputs a sentence, saves it to a file, and then reads it from the file to count the words.

Write a program to determine the number of words in the input sentence.

File Name: sentence\_file.txt

### **Input Format**

The input consists of a single line of text containing words separated by spaces.

### **Output Format**

The output displays the count of words in the sentence.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: Four Words In This Sentence

Output: 5

### **Answer**

```
# You are using Python
sentence = input()
```

```
with open("sentence_file.txt", "w") as file:
    file.write(sentence)
```

```
with open("sentence_file.txt", "r") as file:
    content = file.read()
```

```
word_count = len(content.split())
```

```
print(word_count)
```

**Status :** Correct

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 6\_COD

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

### Section 1 : Coding

#### 1. Problem Statement

A retail store requires a program to calculate the total cost of purchasing a product based on its price and quantity. The program performs validation to ensure valid inputs and handles specific error conditions using exceptions:

Price Validation: If the price is zero or less, raise a ValueError with the message: "Invalid Price". Quantity Validation: If the quantity is zero or less, raise a ValueError with the message: "Invalid Quantity". Cost Threshold: If the total cost exceeds 1000, raise RuntimeError with the message: "Excessive Cost".

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

The first line of input consists of a double value, representing the price of a product.

The second line consists of an integer, representing the quantity of the product.

### **Output Format**

If the calculation is successful, print the total cost rounded to one decimal place.

If the price is zero or less prints "Invalid Price".

If the quantity is zero or less prints "Invalid Quantity".

If the total cost exceeds 1000, prints "Excessive Cost".

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 20.0

5

Output: 100.0

### **Answer**

# You are using Python

```
n=float(input())
```

```
a=int(input())
```

```
d=n*a
```

```
try:
```

```
    if n<=0:
```

```
        raise ValueError("Invalid Price")
```

```
    if a<=0:
```

```
        raise ValueError("Invalid Quantity")
```

```
    if d>1000:
```

```
        raise RuntimeError("Excessive Cost")
```

```
    print(d)
```

```
except ValueError as V:
```

```
    print(V)
```

```
except RuntimeError as r:
```

```
    print(r)
```

**Status : Correct**

**Marks : 10/10**

## 2. Problem Statement

Write a program that calculates the average of a list of integers. The program prompts the user to enter the length of the list (n) and each element of the list. It performs error handling to ensure that the length of the list is a non-negative integer and that each input element is a numeric value.

### ***Input Format***

The first line of the input is an integer n, representing the length of the list as a positive integer.

The second line of the input consists of an element of the list as an integer, separated by a new line.

### ***Output Format***

If the length of the list is not a positive integer or zero, the output displays "Error: The length of the list must be a non-negative integer."

If a non-numeric value is entered for the length of the list, the output displays "Error: You must enter a numeric value."

If a non-numeric value is entered for a list element, the output displays "Error: You must enter a numeric value."

If the inputs are valid, the program calculates and prints the average of the provided list of integers with two decimal places: "The average is: [average]".

Refer to the sample output for the formatting specifications.

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

Input: -2

1

2

Output: Error: The length of the list must be a non-negative integer.

### Answer

# You are using Python

try:

n\_input = input()

n = int(n\_input)

if n < 0 or n > 20:

print("Error: The length of the list must be a non-negative integer.")

else:

numbers = []

for i in range(n):

try:

element = input()

number = int(element)

numbers.append(number)

except ValueError:

print("Error: You must enter a numeric value.")

exit()

if n == 0:

print("The average is: 0.00")

else:

average = sum(numbers) / n

print(f"The average is: {average:.2f}")

except ValueError:

print("Error: You must enter a numeric value.")

**Status : Correct**

**Marks : 10/10**

### 3. Problem Statement

Tara is a content manager who needs to perform case conversions for various pieces of text and save the results in a structured manner.

She requires a program to take a user's input string, save it in a file, and then retrieve and display the string in both upper-case and lower-case versions. Help her achieve this task efficiently.

File Name: text\_file.txt

### ***Input Format***

The input consists of a single line containing a string provided by the user.

### ***Output Format***

The first line displays the original string read from the file in the format: "Original String: {original\_string}".

The second line displays the upper-case version of the original string in the format: "Upper-Case String: {upper\_case\_string}".

The third line displays the lower-case version of the original string in the format: "Lower-Case String: {lower\_case\_string}".

Refer to the sample output for the formatting specifications.

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

Input: #SpecialSymBoLs1234

Output: Original String: #SpecialSymBoLs1234

Upper-Case String: #SPECIALSYMBOLS1234

Lower-Case String: #specialsymbols1234

### ***Answer***

# You are using Python

```
a=input()
```

```
with open ("text.txt","w") as f:
```

```
    f.write(a)
```

```
with open ("text.txt","r") as f:
```

```
    c=f.read()
```

```
print(f"Original String: {c}")
```

```
d=c.upper()
```

```
e=c.lower()
```

```
print(f"Upper-Case String: {d}")
```

```
print(f"Lower-Case String: {e}")
```

**Status :** Correct

**Marks :** 10/10

## **4. Problem Statement**



In a voting system, a person must be at least 18 years old to be eligible to vote. If a user enters an age below 18, the system should raise a user-defined exception indicating that they are not eligible to vote.

### ***Input Format***

The input contains a positive integer representing age.

### ***Output Format***

If the age is less than 18, the output displays "Not eligible to vote".

Otherwise, the output displays "Eligible to vote".

Refer to the sample output for formatting specifications.

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

Input: 18

Output: Eligible to vote

### ***Answer***

# You are using Python

```
class NotEligibleToVote(Exception):  
    pass
```

```
try:  
    age_input = input()  
    age = int(age_input)  
  
    if age < 18:  
        raise NotEligibleToVote  
    else:  
        print("Eligible to vote")
```

```
except NotEligibleToVote:  
    print("Not eligible to vote")  
except ValueError:  
    print("Invalid input. Please enter a numeric value.")
```

Status : Correct

Marks : 10/10

## 5. Problem Statement

Sophie enjoys playing with words and wants to count the number of words in a sentence. She inputs a sentence, saves it to a file, and then reads it from the file to count the words.

Write a program to determine the number of words in the input sentence.

File Name: sentence\_file.txt

### **Input Format**

The input consists of a single line of text containing words separated by spaces.

### **Output Format**

The output displays the count of words in the sentence.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: Four Words In This Sentence

Output: 5

### **Answer**

```
# You are using Python
sentence = input()
```

```
with open("sentence_file.txt", "w") as file:
    file.write(sentence)
```

```
with open("sentence_file.txt", "r") as file:
    content = file.read()
```

```
word_count = len(content.split())
```

```
print(word_count)
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Kishor Kumar Purushothaman  
Email: 240701272@rajalakshmi.edu.in  
Roll no: 240701272  
Phone: 9043229543  
Branch: REC  
Department: I CSE FC  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 6\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

### Section 1 : Coding

#### 1. Problem Statement

In the enchanted realm of Academia, you, the Academic Alchemist, are bestowed with a magical quill and a parchment to weave the grades of aspiring students into a tapestry of academic brilliance.

The mission is to craft a Python program that empowers faculty members to enter student grades for any two subjects, stores these magical grades in a mystical file, and then, with a wave of your virtual wand, calculates the GPA to unveil the true essence of academic achievement.

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

The input format is a string representing the student's name, any two subjects, and corresponding grades.

After entering grades, they can type 'done' when prompted for the student's name.

### **Output Format**

The output should display the (average of grades) calculated GPA with a precision of two decimal places.

The magical grades will be saved in a mystical file named "magical\_grades.txt".

Refer to the sample output for format specifications.

### **Sample Test Case**

Input: Alice

Math

95

English

88

done

Output: 91.50

### **Answer**

# You are using Python

```
file = open("magical_grades.txt", "w")
```

```
while True:
```

```
    name = input()
```

```
    if name.lower() == 'done':
```

```
        break
```

```
    subject1 = input()
```

```
    grade1 = float(input())
```

```
    subject2 = input()
```

```
    grade2 = float(input())
```

```
    file.write(f"{name},{subject1},{grade1},{subject2},{grade2}\n")
```

```
    gpa = (grade1 + grade2) / 2
```

```
    print(f"{gpa:.2f}")
```

```
file.close()
```

**Status : Correct**

**Marks : 10/10**

## 2. Problem Statement

Write a program to obtain the start time and end time for the stage event show. If the user enters a different format other than specified, an exception occurs and the program is interrupted. To avoid that, handle the exception and prompt the user to enter the right format as specified.

Start time and end time should be in the format 'YYYY-MM-DD HH:MM:SS'. If the input is in the above format, print the start time and end time. If the input does not follow the above format, print "Event time is not in the format "

### ***Input Format***

The first line of input consists of the start time of the event.

The second line of the input consists of the end time of the event.

### ***Output Format***

If the input is in the given format, print the start time and end time.

If the input does not follow the given format, print "Event time is not in the format".

Refer to the sample output for formatting specifications.

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

Input: 2022-01-12 06:10:00

2022-02-12 10:10:12

Output: 2022-01-12 06:10:00

2022-02-12 10:10:12

### ***Answer***

```
# You are using Python
from datetime import datetime
```

```
try:
    start_time = input()
    end_time = input()
```

```
start = datetime.strptime(start_time, "%Y-%m-%d %H:%M:%S")
end = datetime.strptime(end_time, "%Y-%m-%d %H:%M:%S")
```

```
print(start_time)
print(end_time)
```

```
except ValueError:
    print("Event time is not in the format")
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

Implement a program that checks whether a set of three input values can form the sides of a valid triangle. The program defines a function `is_valid_triangle` that takes three side lengths as arguments and raises a `ValueError` if any side length is not a positive value. It then checks whether the sum of any two sides is greater than the third side to determine the validity of the triangle.

#### **Input Format**

The first line of input consists of an integer A, representing side1.

The second line of input consists of an integer B, representing side2.

The third line of input consists of an integer C, representing side3.

#### **Output Format**

The output prints either "It's a valid triangle" if the input side lengths form a valid triangle,

or "It's not a valid triangle" if they do not.

If there is a `ValueError`, it should print "ValueError: <error\_message>".

Refer to the sample output for the formatting specifications.

### Sample Test Case

Input: 3

4

5

Output: It's a valid triangle

### Answer

# You are using Python

```
def is_valid_triangle(a, b, c):
```

```
    if a <= 0 or b <= 0 or c <= 0:
```

```
        raise ValueError("Side lengths must be positive")
```

```
    if a + b > c and a + c > b and b + c > a:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
try:
```

```
    side1 = int(input())
```

```
    side2 = int(input())
```

```
    side3 = int(input())
```

```
    if is_valid_triangle(side1, side2, side3):
```

```
        print("It's a valid triangle")
```

```
    else:
```

```
        print("It's not a valid triangle")
```

```
except ValueError as e:
```

```
    print(f"ValueError: {e}")
```

**Status : Correct**

**Marks : 10/10**

### 4. Problem Statement

Alice is developing a program called "Name Sorter" that helps users organize and sort names alphabetically.

The program takes names as input from the user, saves them in a file, and then displays the names in sorted order.

File Name: sorted\_names.txt.



### ***Input Format***

The input consists of multiple lines, each containing a name represented as a string.

To end the input and proceed with sorting, the user can enter 'q'.

### ***Output Format***

The output displays the names in alphabetical order, each name on a new line.

Refer to the sample output for the formatting specifications.

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

Input: Alice Smith

John Doe

Emma Johnson

q

Output: Alice Smith

Emma Johnson

John Doe

### ***Answer***

# You are using Python

```
names = []
```

```
while True:
```

```
    name = input()
```

```
    if name.lower() == 'q':
```

```
        break
```

```
    names.append(name)
```

```
names.sort()
```

```
with open("sorted_names.txt", "w") as file:
```

```
    for name in names:
```

```
        file.write(name + "\n")
```

```
for name in names:
```

```
    print(name)
```

Status : Correct

Marks : 10/10