

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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 : 15

#### Section 1 : MCQ

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

**Answer**

^

**Status :** Correct

**Marks :** 1/1

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

**Answer**

+ operator

**Status :** Correct

**Marks :** 1/1

3. What is the value of the following expression?

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

**Answer**

(1.0,4.0)

**Status :** Correct

**Marks :** 1/1

4. What will be the output for the below code?

```
x=15  
y=12  
print(x&y)
```

**Answer**

12

**Status :** Correct

**Marks :** 1/1

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

**Answer**

int

**Status :** Correct

**Marks :** 1/1

6. Evaluate the expression given below if  $A=16$  and  $B=15$

$A \% B // A$

**Answer**

0

**Status :** Correct

**Marks :** 1/1

7. What is the value of x in the following program?

```
x = int(43.55+2/2)
print(x)
```

**Answer**

44

**Status :** Correct

**Marks :** 1/1

8. Which of the following operators has its associativity from right to left?

**Answer**

\*\*

**Status :** Correct

**Marks :** 1/1

9. Which is the correct operator for power(xy)?

**Answer**

x\*\*y

**Status :** Correct

**Marks :** 1/1

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

**Answer**

Class

**Status :** Correct

**Marks :** 1/1

11. What will be the value of the following Python expression?

4 + 3 % 5

**Answer**

7

**Status :** Correct

**Marks :** 1/1

12. 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

13. What is typecasting in Python?

**Answer**

Change data type property

**Status :** Correct

**Marks :** 1/1

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

**Answer**

float(str)

**Status :** Correct

**Marks :** 1/1

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

```
x = int(34.56 - 2 * 2)  
print(x)
```

**Answer**

30

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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

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

## **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***

```
a=float(input())
b=int(a)
print("The integer value of",a,"is:",b)
```

**Status :** Correct

**Marks :** 1/1

### 3. 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***

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



```
c=a^b  
print("Offer code:",c)
```

**Status :** Correct

**Marks :** 1/1

#### 4. 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**

```
a=int(input())
b=int(input())
c=(50+(a*5))
d=(30+(b*3))
print("Older child allowance: Rs.",c,sep="")
print("Younger child allowance: Rs.",d,sep="")
print("Total allowance: Rs.",c+d,sep="")
```

**Status : Correct**

**Marks : 1/1**

## 5. 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**

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

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

```
print("Initial salaries:\nAlice's salary =",a,"\nBob's salary =",b,"\n\nNew salaries  
after swapping:\nAlice's salary =",b,"\nBob's salary =",a)
```

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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 : 14

#### Section 1 : MCQ

1. 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**

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

```
i = 5
while True:
    if i%10011 == 0:
        break
    print(i, end = " ")
    i += 1
```

**Answer**

5 6 7 8

**Status :** Correct

**Marks :** 1/1

3. Which keyword is used to immediately terminate a loop?

**Answer**

break

**Status :** Correct

**Marks :** 1/1

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

**Answer**

continue

**Status :** Correct

**Marks :** 1/1

5. 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

6. What is the output of the following?

```
for i in range(10):  
    if i == 5:  
        break  
    else:  
        print(i, end=' ')  
else:  
    print("Here")
```

**Answer**

0 1 2 3 4

**Status :** Correct

**Marks :** 1/1

7. 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

8. What is the purpose of the pass statement in Python?

**Answer**

To do nothing and act as a placeholder.

**Status :** Correct

**Marks :** 1/1

9. 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

10. 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

11. 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)
i += 2
```

**Answer**

1 3 5 7 9

**Status :** Correct

**Marks :** 1/1

12. 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

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

**Answer**

When loop condition becomes false

**Status :** Correct

**Marks :** 1/1

14. 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

15. 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 :** Wrong

**Marks :** 0/1

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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**

```
a=int(input())
b=int(input())
f,g,j,l=0,1,0,0
for k in range(a,b+1):
    l+=1
for i in range(0,100):
    if(f>=a and f<=b and f!=g):
        j+=1
    h=g
    g+=f
    f=h
print(l-j)
```

**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**

```
a=int(input())
b=int(input())
c=0
for i in range(a,b+1):
    f=0
    d=i
    h=i
    while(d!=0):
        e=d%10
        f=f*10+e
        d=d//10
    if(f!=i):
        while(h!=0):
            g=h%10
            c+=g
            h=h//10
print(c)
```

**Status : Correct****Marks : 10/10****3. 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**

```
a=int(input())
for i in range(1,a+1):
    if(i%2==0):
        print(i**2)
```

**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 I I W r l d

**Answer**

```
a=input()
for i in a:
    if(i!='a' and i!='A' and i!='e' and i!='E' and i!='i' and i!='l' and i!='o' and i!='O' and i!
    ='u' and i!='U' and ((i>='a' and i<='z') or (i>='A' and i<='Z'))):
        print(i,end=' ')
```

**Status :** Correct

**Marks :** 10/10

**5. 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**

```
a=int(input())
b=int(input())
for i in range(a,b+1):
    c=0
    for j in range(1,i):
        if(i%j==0):
            c+=j
    if(c==i):
        print(i,end=' ')
```

**Status :** Correct

**Marks : 10/10**



# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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 : 23

#### Section 1 : MCQ

1. 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**

2. 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

3. 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

4. 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

5. 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

6. 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**

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

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

**Answer**

oje

**Status : Correct**

**Marks : 1/1**

8. 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**

9. 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**

10. 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

11. 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

12. 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

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**

[25, 14, 222, 33, 2]

**Status :** Wrong

**Marks :** 0/1

15. 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

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

**Answer**

list()

**Status :** Correct

**Marks :** 1/1

17. 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

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

**Answer**

append()

Status : Wrong

Marks : 0/1

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

Answer

Adds a new element to the end of the list

Status : Correct

Marks : 1/1

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

Answer

25

Status : Correct

Marks : 1/1

21. 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

22. 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

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

text = "Python"

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

**Answer**

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

**Status : Correct**

**Marks : 1/1**

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

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

**Answer**

**nohtyP**

**Status : Correct**

**Marks : 1/1**

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

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

**Answer**

**Python**

**Status : Correct**

**Marks : 1/1**

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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

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**

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

```
l=[]
```

```
for i in range(0,a):
```

```
    l.append(int(input()))
```

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

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

```
c=l.pop(b)
```

```
print("List after popping last element:",l,"\nPopped element:",c)
```

**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***

```
a=input()
b=0
print("Area code: ",end=")
for i in a:
    if i==')':
        b=0
    if b==1:
        print(i,end=")
    if i=='(':
```

b=1

**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**

```
a,b=input(),input()
print(a,end="")
c=len(b)
for j in range(c-1,-1,-1):
    print(b[j],end="")
```

**Status :** Correct

**Marks :** 10/10

#### 4. 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***

```
a,b,c=input(),int(input()),int(input())
d=len(a)
if(b<d and c<d and c>b):
    for i in range(b,c+1):
        print(a[i],end="")
```

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

**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

```
input_str = input()
```

```
arr = eval(input_str)
```

```
negatives = []
```

```
positives = []
```

```
for num in arr:
```

```
    if num < 0:
```

```
        negatives.append(num)
```

```
    else:
```

```
        positives.append(num)
```

```
result = negatives + positives
```

```
print("List = ", result)
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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

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):
```

```
    a,b,c,d=0,0,0,0
```

```
    for i in input_string:
```

```
        if i.isalpha():
```

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

```
                a+=1
```

```
            else:
```

```
                b+=1
```

```
        elif i<='9' and i>='0':
```

```
            c+=1
```

```
        else:
```

```
            d+=1
```

```
    return(a,b,c,d)
```

```
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

## 2. 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**

```
a=input()
print(len(a))
```

**Status :** Correct

**Marks :** 10/10

## 3. 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,b=input(),input()
n=lambda s,sub: s.startswith(sub)
print(n(a,b))
```

**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***

```
def armstrong_number(a):
    le,su,temp,d=0,0,int(a),int(a)
    while(temp!=0):
        temp//=10
        le+=1
    for i in range(0,le):
        po=1
        k=d%10
        for j in range(0,le):
            po*=k
        su+=po
        d//=10
    return(su)
x=int(input())
if(x==armstrong_number(x)):
    print(x,"is an Armstrong number.")
```

```
else:  
    print(x,"is not an Armstrong number.")
```

**Status :** Correct

**Marks :** 10/10

## 5. 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  $\text{base}^1$  to  $\text{base}^{\text{exponent}}$  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**

```
a,b,c,d=int(input()),int(input()),list(),1
print(pow(a,b))
for i in range(0,b):
    d*=a
    c.append(d)
print(c,"\n",sum(c))
```

**Status :** Correct

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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 : 18

#### Section 1 : MCQ

1. Which of the following is a Python tuple?

**Answer**

(1, 2, 3)

**Status :** Correct

**Marks :** 1/1

2. 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

3. 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

4. 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

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

**Answer**

()

**Status :** Correct

**Marks :** 1/1

6. What is the output of the following code?

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

```
print(a<b)
```

**Answer**

False

**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. 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

9. 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**

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

**Answer**

Values of a dictionary must be unique

**Status : Correct**

**Marks : 1/1**

11. 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) max(t)
```

**Status : Correct**

**Marks : 1/1**

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

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

**Answer**

13

**Status :** Correct

**Marks :** 1/1

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

**Answer**

Keys must be integers

**Status :** Correct

**Marks :** 1/1

14. 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

15. 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: 'B', 3: 'C'}

**Status :** Correct

**Marks :** 1/1

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

**Answer**

t[3] = 45

**Status :** Correct

**Marks :** 1/1

17. 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

18. What will be the output?

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

**Answer**

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

**Status :** Correct

**Marks :** 1/1

19. 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

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

**Answer**

True

**Status :** Wrong

**Marks :** 0/1

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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

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())
```

```
items = []
```

```
for _ in range(N):
```

```
    # Read the tuple input as a string, then evaluate it to get the actual tuple
```

```
    tup = eval(input())
```

```
    items.append(tup)
```

```
threshold = int(input())
```

```
filtered_quantities = []
```

```
for item in items:
```

```
    # item is a tuple: (ID, [quantities])
```

```
quantities = item[1]
# Use filter to get quantities greater than threshold
filtered = filter(lambda x: x > threshold, quantities)
filtered_quantities.extend(filtered)

print(*filtered_quantities)
```

**Status :** Correct

**Marks :** 10/10

## 2. 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())
```

```
freq = {}
```

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

```
    product_id = int(input())
```

```
    freq[product_id] = freq.get(product_id, 0) + 1
```

```
total_unique = len(freq)
```

```
average_freq = sum(freq.values()) / total_unique
```

```
print(freq)
```

```
print("Total Unique IDs:", total_unique)
```

```
print(f"Average Frequency: {average_freq:.2f}")
```

**Status : Correct**

**Marks : 10/10**

### **3. 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())
```

```
result = {}
```

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

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

```
    customer_id = data[0]
```

```
    amounts = data[1:]
```

```
    total = sum(amounts)
```

```
maximum = max(amounts)
result[customer_id] = [total, maximum]

print(result)
```

**Status :** Correct

**Marks :** 10/10

#### 4. 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

```
registered = set(map(int, input().split()))
```

```
attended = set(map(int, input().split()))
```

```
dropped_out = set(map(int, input().split()))
```

```
registered_and_attended = registered & attended
```

```
attended_not_dropped = registered_and_attended - dropped_out
```

```
print(registered_and_attended)
```

```
print(attended_not_dropped)
```

**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())
```

```
list1 = list(map(int, input().split(',')))
```

```
list2 = list(map(int, input().split(',')))
```

```
result = tuple(a + b for a, b in zip(list1, list2))
```

```
print(result)
```

**Status : Correct**

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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 : 15

#### Section 1 : MCQ

1. 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

2. 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

3. How do you create a user-defined exception in Python?

**Answer**

By using the raise keyword

**Status :** Wrong

**Marks :** 0/1

4. 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

5. What happens if an exception is not caught in the except clause?

**Answer**

```
raise CustomException()
```

**Status :** Wrong

**Marks :** 0/1

6. What is the correct way to raise an exception in Python?

**Answer**

```
raise Exception()
```

**Status :** Correct

**Marks :** 1/1

7. 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

8. 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

9. 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

10. What happens if no arguments are passed to the seek function?

**Answer**

file position is set to the end of file

**Status :** Wrong

**Marks :** 0/1

11. 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



12. 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

13. 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

14. 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

15. What is the purpose of the except clause in Python?

**Answer**

To handle exceptions during code 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

Ouputput:

['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 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

18. Which clause is used to clean up resources, such as closing files in Python?

**Answer**

finally

**Status :** Correct

**Marks :** 1/1

19. How do you rename a file?

**Answer**

os.rename(existing\_name, new\_name)

**Status :** Correct

**Marks :** 1/1

20. Which of the following is true about fp.seek(10,1)

**Answer**

Move file pointer ten characters ahead from the beginning of a file

**Status :** Wrong

**Marks :** 0/1

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
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

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:

        print("Error: The length of the list must be a non-negative integer.")

    elif n > 20:

        print("Error: The length of the list must be a non-negative integer.")

    else:

        numbers = []

        for \_ in range(n):

            element\_input = input()

            try:

                number = int(element\_input)

                numbers.append(number)

```
except ValueError:
    print("Error: You must enter a numeric value.")
    exit()
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

## 2. 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

try:

price\_input = input()

quantity\_input = input()

price = float(price\_input)

quantity = int(quantity\_input)

if price <= 0:

raise ValueError("Invalid Price")

if quantity <= 0:

raise ValueError("Invalid Quantity")

total\_cost = price \* quantity

if total\_cost > 1000:

raise RuntimeError("Excessive Cost")

print(f"{total\_cost:.1f}")

except ValueError as ve:

print(ve)

except RuntimeError as re:

print(re)

**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
```

```
# Read input from the user
```

```
input_string = input()
```

```
# Write the input string to the file  
with open("text_file.txt", "w") as file:  
    file.write(input_string)
```

```
# Read the string back from the file
```



```
with open("text_file.txt", "r") as file:  
    original_string = file.read()
```

```
# Convert the string to upper-case and lower-case  
upper_case_string = original_string.upper()  
lower_case_string = original_string.lower()
```

```
# Print the results in the specified format  
print(f"Original String: {original_string}")  
print(f"Upper-Case String: {upper_case_string}")  
print(f"Lower-Case String: {lower_case_string}")
```

**Status :** Correct

**Marks : 10/10**

#### 4. 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
# Read the input sentence from the user
sentence = input()

# Write the sentence to a file
with open("sentence_file.txt", "w") as file:
    file.write(sentence)

# Read the sentence back from the file
with open("sentence_file.txt", "r") as file:
    content = file.read()

# Split the sentence into words and count them
word_count = len(content.split())

# Print the word count
print(word_count)
```

**Status :** Correct

**Marks : 10/10**

## 5. 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
# Define a user-defined exception
class NotEligibleToVote(Exception):
    pass

# Read the age input
age = int(input())

try:
    if age < 18:
        # Raise custom exception if age is less than 18
        raise NotEligibleToVote
    else:
        print("Eligible to vote")
except NotEligibleToVote:
    print("Not eligible to vote")
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 7\_MCQ

Attempt : 1  
Total Mark : 20  
Marks Obtained : 15

#### Section 1 : MCQ

1. What does NumPy stand for?

**Answer**

Numerical Python

**Status :** Correct

**Marks :** 1/1

2. Which NumPy function is used to create an identity matrix?

**Answer**

numpy.eye()

**Status :** Wrong

**Marks :** 0/1

3. Minimum number of argument we require to pass in pandas series ?

**Answer**

0

**Status : Wrong**

**Marks : 0/1**

4. What is the output of the following NumPy code snippet?

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
r = arr[arr > 2]
print(r)
```

**Answer**

[3 4 5]

**Status : Correct**

**Marks : 1/1**

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

```
import numpy as np
arr = np.array([1, 2, 3])
result = np.concatenate((arr, arr))
print(result)
```

**Answer**

[1 2 3 1 2 3]

**Status : Correct**

**Marks : 1/1**

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

```
import pandas as pnd
pnd.Series([1,2], index= ['a','b','c'])
```

**Answer**

Value Error

**Status :** Correct

**Marks :** 1/1

7. Which NumPy function is used to find the indices of the maximum and minimum values in an array?

**Answer**

argmax() and argmin()

**Status :** Correct

**Marks :** 1/1

8. What is the purpose of the following NumPy code snippet?

```
import numpy as np
arr = np.zeros((3, 4))
print(arr)
```

**Answer**

Displays a 3x4 matrix filled with zeros

**Status :** Correct

**Marks :** 1/1

9. What is the primary data structure used in NumPy for numerical computations?

**Answer**

Array

**Status :** Correct

**Marks :** 1/1

10. What is the output of the following NumPy code?

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
r = arr[2:4]
print(r)
```

**Answer**

[3 4]

**Status :** Correct

**Marks :** 1/1

11. In NumPy, how do you access the first element of a one-dimensional array arr?

**Answer**

arr[0]

**Status :** Correct

**Marks :** 1/1

12. What does the np.arange(10) function in NumPy do?

**Answer**

Creates an array with values from 0 to 10

**Status :** Wrong

**Marks :** 0/1

13. What is the primary purpose of Pandas DataFrame?

**Answer**

To store data in tabular form for analysis and manipulation

**Status :** Correct

**Marks :** 1/1

14. The important data structure of pandas is/are \_\_\_\_.

**Answer**

Both Series and Data Frame

**Status :** Correct

**Marks :** 1/1

15. Which of the following is a valid way to import NumPy in Python?

**Answer**

```
import numpy as np
```

**Status :** Correct

**Marks :** 1/1

16. What is the output of the following code?

```
import numpy as np
a = np.arange(10)
print(a[2:5])
```

**Answer**

[2, 3, 4]

**Status :** Correct

**Marks :** 1/1

17. In the DataFrame created in the code, what is the index for the row containing the data for 'Jack'?

```
import pandas as pd
```

```
data = {'Name': ['Tom', 'Jack', 'nick', 'juli'],
        'marks': [99, 98, 95, 90]}
```

```
df = pd.DataFrame(data, index=['rank1',
                                'rank2',
                                'rank3',
                                'rank4'])
```

```
print(df)
```

**Answer**

rank1

**Status :** Wrong

**Marks :** 0/1

18. Which NumPy function is used to calculate the standard deviation of an array?

**Answer**



`numpy.std()`

**Status :** Correct

**Marks :** 1/1

19. What is the result of the following NumPy operation?

```
import numpy as np
arr = np.array([1, 2, 3])
r = arr + 5
print(r)
```

**Answer**

[1 2 3 5]

**Status :** Wrong

**Marks :** 0/1

20. Which function is used to create a Pandas DataFrame?

**Answer**

`pd.DataFrame()`

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Natarajan S  
Email: 240701347@rajalakshmi.edu.in  
Roll no: 2116240701347  
Phone: 8122886273  
Branch: REC  
Department: I CSE AH  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 7\_COD

Attempt : 1  
Total Mark : 50  
Marks Obtained : 50

### Section 1 : Coding

#### 1. Problem Statement

Sita is analyzing her company's daily sales data to find all sales values that are multiples of 5 and exceed 100. She wants to filter these specific sales values from the list.

Help her to implement the task using the numpy package.

Formula:

To filter sales values:

Select all values  $s$  from sales such that  $(s \% 5 == 0)$  and  $(s > 100)$

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

The first line of input consists of an integer value,  $n$ , representing the number of

sales entries.

The second line of input consists of n floating-point values, sales, separated by spaces, representing daily sales figures.

### **Output Format**

The output prints: filtered\_sales

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 5

50.0 100.0 105.0 150.0 99.0

Output: [105. 150.]

### **Answer**

```
import numpy as np
```

```
# Read input
```

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

```
sales = np.array(list(map(float, input().split())))
```

```
# Filter: multiples of 5 and greater than 100
```

```
filtered_sales = sales[(sales % 5 == 0) & (sales > 100)]
```

```
# Print output in correct format
```

```
print(filtered_sales)
```

**Status :** Correct

**Marks :** 10/10

## **2. Problem Statement**

A company tracks the monthly sales data of various products. You are given a table where each row represents a product and each column represents its monthly sales in sequential months.

Your task is to compute the cumulative monthly sales for each product

using numpy, where the cumulative sales for a month is the total sales from month 1 up to that month.

### ***Input Format***

The first line of input consists of two integer values, products and months, separated by a space.

Each of the next products lines consists of months integer values representing the monthly sales data of a product.

### ***Output Format***

The first line of output prints: "Cumulative Monthly Sales:"

The second line of output prints: the 2D numpy array cumulative\_array that contains the cumulative sales data for each product.

Refer to the sample output for the formatting specifications.

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

Input: 2 4

10 20 30 40

5 15 25 35

Output: Cumulative Monthly Sales:

[[ 10 30 60 100]

[ 5 20 45 80]]

### ***Answer***

```
import numpy as np
```

```
# Read input: number of products and months
products, months = map(int, input().split())
```

```
# Read sales data for each product
sales_data = []
for _ in range(products):
    row = list(map(int, input().split()))
    sales_data.append(row)
```

```
# Convert to NumPy array
sales_array = np.array(sales_data)

# Compute cumulative sum along each row (axis=1)
cumulative_array = np.cumsum(sales_array, axis=1)

# Output as specified
print("Cumulative Monthly Sales:")
print(cumulative_array)
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

Rekha works in hospital data management and receives patient records with missing or incomplete data. She needs to clean the records by performing the following tasks:

Calculate the mean of the available Age values. Replace any missing (NaN) values in the Age column with this mean age. Remove any rows where the Diagnosis value is missing (NaN). Reset the DataFrame index after removing these rows.

Implement this data cleaning task using the pandas package.

#### **Input Format**

The first line of input contains an integer  $n$  representing the number of patient records.

The second line contains the CSV header — comma-separated column names (e.g., "Name, Age, Diagnosis, Gender").

The next  $n$  lines each contain one patient record in comma-separated format.

#### **Output Format**

The first line of output is the text:

Cleaned Hospital Records:

The next lines print the cleaned pandas DataFrame (as produced by `print(cleaned_df)`).

This will include the updated values of the Age column (with missing ages filled by the mean age), and any rows with missing Diagnosis removed.

The DataFrame will be displayed using the default pandas print() representation.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 5

PatientID,Name,Age,Diagnosis

1,John Doe,45,Flu

2,Jane Smith,,Cold

3,Bob Lee,50,

4,Alice Green,38,Fever

5,Tom Brown,,Infection

Output: Cleaned Hospital Records:

	PatientID	Name	Age	Diagnosis
0	1	John Doe	45.000000	Flu
1	2	Jane Smith	44.333333	Cold
2	4	Alice Green	38.000000	Fever
3	5	Tom Brown	44.333333	Infection

### **Answer**

```
import pandas as pd
import sys
```

```
# Read input
n = int(input())
header = input().strip().split(',')
```

```
# Read n lines of patient records
data = [input().strip().split(',') for _ in range(n)]
```

```
# Create DataFrame
df = pd.DataFrame(data, columns=header)
```

```
# Convert Age column to numeric (to handle NaN properly)
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
```

```
# Calculate mean age (ignoring NaN)
mean_age = df['Age'].mean()

# Fill missing Age values with mean age
df['Age'] = df['Age'].fillna(mean_age)

# Drop rows where Diagnosis is missing (NaN)
df = df[df['Diagnosis'].notna() & (df['Diagnosis'] != "")]

# Reset index
cleaned_df = df.reset_index(drop=True)

# Output
print("Cleaned Hospital Records:")
print(cleaned_df)
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement

Alex is a data scientist analyzing the relationship between two financial indicators over time. He has collected two time series datasets representing daily values of these indicators over several months. Alex wants to understand how these two indicators correlate at different time lags to identify possible leading or lagging behaviors.

Your task is to help Alex compute the cross-correlation of these two time series using numpy, so he can analyze the similarity between the two signals at various time shifts.

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

The first line of input consists of space-separated float values representing the first time series, array1.

The second line of input consists of space-separated float values representing the second time series, array2.

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

The first line of output prints: "Cross-correlation of the two time series:"

The second line of output prints: the 1D numpy array `cross_corr` representing the cross-correlation of `array1` and `array2` across different lags.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 1.0 2.0 3.0  
4.0 5.0 6.0

Output: Cross-correlation of the two time series:  
[ 6. 17. 32. 23. 12.]

### **Answer**

```
import numpy as np

# Read input
array1 = np.array(list(map(float, input().split())))
array2 = np.array(list(map(float, input().split())))

# Compute cross-correlation
cross_corr = np.correlate(array1, array2, mode='full')

# Output result
print("Cross-correlation of the two time series:")
print(cross_corr)
```

**Status :** Correct

**Marks :** 10/10

## **5. Problem Statement**

Sita works as a sales analyst and needs to analyze monthly sales data for different cities. She receives lists of cities, months, and corresponding sales values and wants to create a pandas DataFrame using a MultiIndex of cities and months.

Help her to implement this task and calculate total sales for each city.



### ***Input Format***

The first line of input consists of an integer value,  $n$ , representing the number of records.

The second line of input consists of  $n$  space-separated city names.

The third line of input consists of  $n$  space-separated month names.

The fourth line of input consists of  $n$  space-separated float values representing sales for each city-month combination.

### ***Output Format***

The first line of output prints: "Monthly Sales Data with MultiIndex:"

The next lines print the DataFrame with MultiIndex (City, Month) and their corresponding sales values.

The following line prints: "\nTotal Sales Per City:"

The final lines print the total sales per city, computed by grouping the sales data on city names.

Refer to the sample output for the formatting specifications.

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

Input: 4

NYC NYC LA LA

Jan Feb Jan Feb

100 200 300 400

Output: Monthly Sales Data with MultiIndex:

Sales

City Month

NYC Jan 100.0

Feb 200.0

LA Jan 300.0

Feb 400.0

Total Sales Per City:

```
Sales
City
LA    700.0
NYC   300.0
```

**Answer**

```
import pandas as pd

# Read input
n = int(input())
cities = input().split()
months = input().split()
sales = list(map(float, input().split()))

# Create MultiIndex from city and month
index = pd.MultiIndex.from_arrays([cities, months], names=['City', 'Month'])

# Create DataFrame
df = pd.DataFrame({'Sales': sales}, index=index)

# Compute total sales per city
total_sales = df.groupby('City').sum()

# Output
print("Monthly Sales Data with MultiIndex:")
print(df)
print("\nTotal Sales Per City:")
print(total_sales)
```

**Status :** Correct

**Marks :** 10/10