

# Rajalakshmi Engineering College

Name: ZARAA JUHI SAAI K  
Email: 240701044@rajalakshmi.edu.in  
Roll no: 240701044  
Phone: 9444513456  
Branch: REC  
Department: I CSE FA  
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 : 21

#### Section 1 : MCQ

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

```
txt = "My Classroom"  
print(txt.find("o"))  
print(txt.index("o"))
```

**Answer**

Compile Time Error

**Status : Wrong**

**Marks : 0/1**

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

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

```
print(answer)
```

**Answer**

Python

**Status :** Correct

**Marks :** 1/1

3. 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

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

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

**Answer**

5

**Status :** Wrong

**Marks :** 0/1

5. 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

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

**Answer**

25

**Status :** Correct

**Marks :** 1/1

7. 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

8. 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

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

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

**Answer**

oje

**Status :** Correct

**Marks :** 1/1

10. 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

11. 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

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

**Answer**

extend()

**Status :** Correct

**Marks :** 1/1

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

**Answer**

Adds a new element to the end of the list

**Status :** Correct

**Marks :** 1/1

14. 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

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. 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

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

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

**Answer**

nohtyP

**Status :** Correct

**Marks :** 1/1

18. 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

19. 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

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

**Answer**

empty\_list()

**Status :** Wrong

**Marks :** 0/1

21. 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

22. 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

23. 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

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

**Answer**

25

**Status :** Wrong

**Marks :** 0/1

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

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

**Answer**

\*\*Python\*\*

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: ZARAA JUHI SAAI K  
Email: 240701044@rajalakshmi.edu.in  
Roll no: 240701044  
Phone: 9444513456  
Branch: REC  
Department: I CSE FA  
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 : 40

### Section 1 : Coding

#### 1. 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=[x for x in arr if x<0]
non_negatives=[x for x in arr if x>=0]
result=negatives+non_negatives

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

**Status :** Correct

**Marks :** 10/10

## **2. 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***

```
str1=input()
str2=input()
print(str1+str2[::-1])
```

**Status :** Correct

**Marks :** 10/10

## **3. 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
s=input()
print(f"Area code:{s[1:4]}")
```

**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

```
def manage_list():
```

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

```
    elements = []
```

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

```
        elements = int(input())
```

```
        elements.append(element)
```

```
M=int(input())
```

```
print(f"List after appending elements:{elements}")
```

```
popped_element=element.pop(M)

print(f"List after popping list elements:{elements}")

print(f"Popped element:{popped_element}")
manage_list()
```

**Status : Wrong**

**Marks : 0/10**

## 5. 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
```

```
s=input()
```

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

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

```
if(m<=n or n>len(s)):
```

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

```
else:
```

```
    print(s[n:m+1])
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: ZARAA JUHI SAAI K  
Email: 240701044@rajalakshmi.edu.in  
Roll no: 240701044  
Phone: 9444513456  
Branch: REC  
Department: I CSE FA  
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())

l=list(map(int,input().split()))

p=[]

for j in l:

if j>0:

p.append(j)

p.sort()

def idk\_what\_to\_name\_this\_function\_lol():

mi=1

for i in p:

if i==mi:

mi+=1

elif i>mi:

break

print(mi)

idk\_what\_to\_name\_this\_function\_lol()

#def birds\_are\_government\_drones():

# mi=p[0]#1

# mx=p[len(p)-1]#4

# if mi>1:

# print(1)

# return

# t=[i for i in range(mi,mx+2)]

# # print(t,"is the fun array")

# # print(p,"is the sorted array")

# for i in range(mi,mx+!):

# if t[i] not in p:

# print(t[i])



# return

**Status :** Correct

**Marks :** 10/10

## 2. 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())
def AbCdEfGhIjKlMnOpQrStUvWxYz():
    l=[]
    for i in range(n):
        t=int(input())
        l.append(t)
    l.sort()
    print(f"Original list: {l}")
    t=[l[i]**3 if i%2==0 else l[i]**2 for i in range(1,len(l))]
    t.insert(0,0)
    print(f"Replaced list: {t}")

AbCdEfGhIjKlMnOpQrStUvWxYz()
```

**Status :** Correct

**Marks : 10/10**

### 3. 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
l=input().strip()
s=input()
def gowri_has_a_skill_issue():
    print(l.count(s))
gowri_has_a_skill_issue()
```

**Status :** Correct

**Marks :** 10/10

## **4. 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

```
s=input()
```

```
def question_too_small_so_i_beefed_it_up_with_functions():
```

```
    print(f"Length of the string: {len(s)}")
```

```
    print(f"Copied string: {s}")
```

```
    return
```

```
question_too_small_so_i_beefed_it_up_with_functions()
```

**Status :** Correct

**Marks :** 10/10

## **5. 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

l = list(map(int,input().split()))

p=[]

def the\_only\_thing\_that\_is\_peak\_is\_onepiece():

for i in range(len(l)):

if i==len(l)-1:

if l[i]>l[i-1]:

```
        p.append(l[i])
    elif l[i]>l[i-1] and l[i]>l[i+1]:
        p.append(l[i])

    print(f"Peaks: {p}")
the_only_thing_that_is_peak_is_onepiece()
```

**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

```
s=input()
```

```
def abcdefghijklmnopqrstuvwxyz():
```

```
    upper=0
```

```
    lower=0
```

```
    digit=0
```

```
    special=0
```

```
    for i in s:
```

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

```
            upper+=1
```

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

```
            lower+=1
```

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

```
            digit+=1
```

```
        elif not i.isalnum():
```

```
            special+=1
```

```
print(f"Uppercase letters: {upper}")
```

```
print(f"Lowercase letters: {lower}")
```

```
print(f"Digits: {digit}")
```

```
print(f"Special characters: {special}")
```

```
abcdefghijklmnopqrstuvwxyz()
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: ZARAA JUHI SAAI K  
Email: 240701044@rajalakshmi.edu.in  
Roll no: 240701044  
Phone: 9444513456  
Branch: REC  
Department: I CSE FA  
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 : 30

### Section 1 : Coding

#### 1. 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**

```
# You are using Python
initial_list=list(map(int,input().strip().split()))
print("List1:",initial_list)

remove_item=int(input().strip())

if remove_item in initial_list:
    initial_list.remove(remove_item)
    print("List after removal:",initial_list)
else:
    print("Element not found in the list")

new_items = list(map(int,input().strip().split()))
initial_list.extend(new_items)

print("Final list:",initial_list)
```

**Status :** Correct

**Marks :** 10/10

## 2. 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

```
def is_palindrome(n):
```

```
    return str(n) == str(n)[::-1]
```

```
def process_message(s):
```

```
    digit_sum=sum(int(char) for char in s if char.isdigit())
```

```
    output = f"{digit_sum}"
```

```
    if digit_sum>9:
```

```
        if is_palindrome(digit_sum):
```

```
            output+= "Palindrome"
```

```
        else:
```

```
            output+= "Not palindrome"
```

```
    print(output)
```

# Example usage:

```
message=input()
```

```
process_message(message)
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement

A company is creating email accounts for its new employees. They want to use a naming convention for email addresses that consists of the first letter of the employee's first name, followed by their last name, followed by @company.com.

The company also has a separate email domain for administrative employees.

Write a program that prompts the user for their first name, last name, role, and company and then generates their email address using the appropriate naming convention based on their role. This is demonstrated in the below examples.

**Note:**

The generated email address should consist of the first letter of the first name, the last name in lowercase, and a suffix based on the role and company, all in lowercase.

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

The first line of input consists of the first name of an employee as a string.

The second line consists of the last name of an employee as a string.

The third line consists of the role of the employee as a string.

The last line consists of the company name as a string.

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

The output consists of a single line containing the generated email address for the employee, following the specified naming convention.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: John

Smith

admin

iamNeo

Output: jsmith@admin.iamneo.com

### **Answer**

# You are using Python

```
first_name=input().strip()
```

```
last_name=input().strip()
```

```
role=input().strip()
```

```
company=input().strip()
```

```
email_prefix=first_name[0].lower()+last_name.lower()
```

```
if role.lower() == 'admin':
```

```
    domain=f"admin.{company.lower()}.com"
```

```
else:
```

```
    domain=f"{company.lower()}.com"
```

```
email=f"{email_prefix}@{domain}"
```

```
print(email)
```

**Status : Correct**

**Marks : 10/10**