

# Rajalakshmi Engineering College

Name: nila ela  
Email: 241901074@rajalakshmi.edu.in  
Roll no: 241901074  
Phone: 9025155667  
Branch: REC  
Department: I CSE (CS) FB  
Batch: 2028  
Degree: B.E - CSE (CS)

Scan to verify results



## NeoColab\_REC\_CS23221\_Python Programming

### REC\_Python\_Week 3\_MCQ

Attempt : 1  
Total Mark : 25  
Marks Obtained : 24

#### Section 1 : MCQ

1. 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**

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

**Answer**

[2, 33, 222, 14]

**Status :** Correct

**Marks :** 1/1

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

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

**Answer**

3

**Status :** Correct

**Marks :** 1/1

4. 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

5. 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

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

**Answer**

`extend()`

**Status :** Correct

**Marks :** 1/1

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

**Answer**

`list()`

**Status :** Correct

**Marks :** 1/1

8. 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

9. 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

10. 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

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 is the output of the following Python code?

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

**Answer**

nohtyP

**Status :** Correct

**Marks :** 1/1

13. 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

14. 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

15. 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

16. 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, 5]

**Status :** Wrong

**Marks :** 0/1

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

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

**Answer**

\*\*Python\*\*

**Status :** Correct

**Marks :** 1/1

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

**Answer**

25

**Status : Correct**

**Marks : 1/1**

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

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

**Answer**

2

**Status : Correct**

**Marks : 1/1**

20. 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**

21. 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**

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

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

**Answer**

oje

**Status : Correct**

**Marks : 1/1**

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

**Answer**

Adds a new element to the end of the list

**Status : Correct**

**Marks : 1/1**

24. 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**

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: nila ela  
Email: 241901074@rajalakshmi.edu.in  
Roll no: 241901074  
Phone: 9025155667  
Branch: REC  
Department: I CSE (CS) FB  
Batch: 2028  
Degree: B.E - CSE (CS)

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**

# You are using Python

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

```
elements = []
```

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

```
    num = int(input())
```

```
    elements.append(num)
```

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

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

```
popped_element = elements.pop(M)
```

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

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

**Status :** Correct

**Marks :** 10/10

## 2. 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
inputl = input()
arr = list(map(int, inputl.strip('[]').split(',')))
negatives = [x for x in arr if x < 0]
positives = [x for x in arr if x >= 0]
result = negatives + positives
print("List = ", result)
```

**Status :** Correct

**Marks :** 10/10

## **3. Problem Statement**

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

Ram needs your help to design a program.

### **Input Format**

The input consists of two strings in separate lines.

### **Output Format**

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

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: hello  
word

Output: hellodrow

### **Answer**

```
# You are using Python
s1=input()
s2=input()
s3=s2[::-1]
print(s1+s3)
```

**Status :** Correct

**Marks : 10/10**

## **4. 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
s1=input()
s2=s1[1:4]
print("Area code:",s2)
```

**Status :** Correct

**Marks : 10/10**

### **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
```

```
n=input()
```

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

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

```
if (0 <= start <= end < len(n)):
```

```
    print(n[start:end+1])
```

```
else:
```

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

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: nila ela  
Email: 241901074@rajalakshmi.edu.in  
Roll no: 241901074  
Phone: 9025155667  
Branch: REC  
Department: I CSE (CS) FB  
Batch: 2028  
Degree: B.E - CSE (CS)

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
def find_missing_positive(n, arr):
    for i in range(n):
        while 1 <= arr[i] <= n and arr[arr[i] - 1] != arr[i]:
            arr[arr[i] - 1], arr[i] = arr[i], arr[arr[i] - 1]

    for i in range(n):
        if arr[i] != i + 1:
            return i + 1
    return n + 1

n = int(input())
arr = list(map(int, input().split()))

print(find_missing_positive(n, arr))
```

**Status :** Correct

**Marks : 10/10**

**2. Problem Statement**

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

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

Help Neha implement a program that efficiently performs these operations.

**Input Format**

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



### **Output Format**

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

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

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: technology-23

Output: Length of the string: 13

Copied string: technology-23

### **Answer**

```
# You are using Python
def string_operations(A):
```

```
    length_of_string = len(A)
    B = A[:]
    print(f"Length of the string: {length_of_string}")
    print(f"Copied string: {B}")
```

```
A = input().strip()
```

```
string_operations(A)
```

**Status :** Correct

**Marks :** 10/10

### **3. 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

```
def transform_list(n, numbers):
```

```
    numbers.sort()
```

```
    replaced_list = []
```

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

```
        if i == 0:
```

```
            replaced_list.append(0)
```

```
        elif i % 2 == 0:
```

```
        replaced_list.append(numbers[i] ** 3)
    else:
        replaced_list.append(numbers[i] ** 2)

    print(f"Original List: {numbers}")
    print(f"Replaced List: {replaced_list}")

N = int(input())
numbers = [int(input()) for _ in range(N)]

transform_list(N, numbers)
```

**Status :** Correct

**Marks : 10/10**

#### 4. Problem Statement

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

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

Note: Case-sensitive

##### **Input Format**

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

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

##### **Output Format**

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

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

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

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

Output: 3

### **Answer**

```
# You are using Python
import re
```

```
def count_word_occurrences(text, word):
    pattern = r'\b' + re.escape(word) + r'\b'
    matches = re.findall(pattern, text)
    return len(matches)
```

```
text = input()
word = input()
```

```
print(count_word_occurrences(text, word))
```

**Status :** Correct

**Marks : 10/10**

## **5. 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

```
def analyze_log_entry(log_entry):
```

```
    uppercase_count = 0
```

```
    lowercase_count = 0
```

```
    digits_count = 0
```

```
    special_count = 0
```

```
    for char in log_entry:
```

```
        if char.isupper():
```

```
            uppercase_count += 1
```

```
        elif char.islower():
```

```
            lowercase_count += 1
```

```
        elif char.isdigit():
```

```
            digits_count += 1
```

```
        else:
```

```
            special_count += 1
```

```
print(f"Uppercase letters: {uppercase_count}")
print(f"Lowercase letters: {lowercase_count}")
print(f"Digits: {digits_count}")
print(f"Special characters: {special_count}")
```

```
log_entry=input()
analyze_log_entry(log_entry)
```

**Status :** Correct

**Marks :** 10/10

## 6. 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

```
measurements = [int(x) for x in input().split()]
```

```
peaks = [
```

```
    measurements[i]
```

```
    for i in range(len(measurements))
```

```
    if (i == 0 and (len(measurements) == 1 or measurements[i] > measurements[i
```

```
+1])) or
```

```
    (i > 0 and i < len(measurements) - 1 and measurements[i] > measurements[i -
```

```
1] and measurements[i] > measurements[i + 1]) or
```

```
    (i == len(measurements) - 1 and len(measurements) > 1 and
```

```
measurements[i] > measurements[i - 1])
```

```
]
```

```
print(f"Peaks: {peaks}")
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: nila ela

Email: 241901074@rajalakshmi.edu.in

Roll no: 241901074

Phone: 9025155667

Branch: REC

Department: I CSE (CS) FB

Batch: 2028

Degree: B.E - CSE (CS)

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_input = input().strip()
shopping_list = initial_input.split()

print(f"List1: [{', '.join(shopping_list)}]")

item_to_remove = input().strip()

if item_to_remove in shopping_list:
    shopping_list.remove(item_to_remove)
    print(f"List after removal: [{', '.join(shopping_list)}]")
else:
    print("Element not found in the list")

new_items = input().strip().split()

shopping_list.extend(new_items)

print(f"Final list: [{', '.join(shopping_list)}]")
```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

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

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

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

### ***Input Format***

The input consists of two strings in each line.

### ***Output Format***

The output displays the concatenated string in the mentioned format.

Refer to the sample output for formatting specifications.

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

Input: abc  
def

Output: adbecf

### ***Answer***

```
# You are using Python
str1 = input().strip()
str2 = input().strip()
```

```
result = []
```

```
for i in range(len(str1)):
    result.append(str1[i])
    result.append(str2[i])
```

```
print("".join(result))
```

**Status :** Correct

**Marks :** 10/10

## **3. Problem Statement**

Raj wants to write a program that takes a list of strings as input and returns the longest word in the list. If there are multiple words with the same length, the program should return the first one encountered.

Help Raj in his task.

### ***Input Format***

The input consists of a single line of space-separated strings.

### ***Output Format***

The output prints a string representing the longest word in the given list.

Refer to the sample output for formatting specifications.

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

Input: cat dog elephant lion tiger giraffe

Output: elephant

### ***Answer***

```
# You are using Python
def find_longest_word():
```

```
    input_string = input().strip()
```

```
    words = input_string.split()
```

```
    longest_word = words[0]
```

```
    for word in words:
```

```
        if len(word) > len(longest_word):
            longest_word = word
```

```
    print(longest_word)
```

```
find_longest_word()
```

**Status :** Correct

**Marks :** 10/10