

Rajalakshmi Engineering College

Name: Akhil Kumaran
Email: 240701026@rajalakshmi.edu.in
Roll no: 240701026
Phone: 8838850965
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 : 24

Section 1 : MCQ

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

Answer

extend()

Status : Correct

Marks : 1/1

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

Answer

Adds a new element to the end of the list

Status : Correct

Marks : 1/1

3. What 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

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. 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

7. 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

8. 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

9. 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

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

Answer

The first three elements of the list

Status : Wrong

Marks : 0/1

11. 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

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

Answer

list()

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

Answer

25

Status : Correct

Marks : 1/1

15. 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

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

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

Answer

Python

Status : Correct

Marks : 1/1

17. 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

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

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

Answer

oje

Status : Correct

Marks : 1/1

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

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

Answer

nohtyP

Status : Correct

Marks : 1/1

20. 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

21. 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

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

Answer

[2, 33, 222, 14]

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. 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

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

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

Answer

3

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: Akhil Kumaran
Email: 240701026@rajalakshmi.edu.in
Roll no: 240701026
Phone: 8838850965
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 : 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())
```

```
s=[]
```

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

```
    g=int(input())
```

```
    s+=[g]
```

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

```
index=int(input())
```

```
a=s.pop(index)
```

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

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

Status : Correct

Marks : 10/10

2. Problem Statement

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

Write a Python program for the same.

Note

(XXX) - Area code

XXX-XXXX - Phone number

Input Format

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

Output Format

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: (123) 456-7890

Output: Area code: 123

Answer

```
# You are using Python
s1=input()
print("Area code:",s1[1:4:])
```

Status : Correct

Marks : 10/10

3. 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

```
input_string=input().strip()
start=int(input())
end=int(input())
if 0<=start <=end <len(input_string):
    print(input_string[start:end+1])
```

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

Status : Correct

Marks : 10/10

4. 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

```
n=str(input())  
m=str(input())  
u=""  
index=-1  
for i in m:  
    u+=m[index]  
    index-=1  
print(n+u)
```

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

```
input_list = input().strip()
numbers = list(map(int,input_list.strip('[]').split(',')))
negatives = [num for num in numbers if num<0]
positives = [num for num in numbers if num>=0]
result = negatives + positives
print("List=",result)
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Akhil Kumaran
Email: 240701026@rajalakshmi.edu.in
Roll no: 240701026
Phone: 8838850965
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 : 57.5

Section 1 : Coding

1. 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

```
a=input()
length = len(a)
b=a
print("Length of the string:",length)
print("Copied string:",b,end="")
```

Status : Correct

Marks : 10/10

2. 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

```
log_entry=input()
uppercase=0
lowercase=0
digits=0
special=0
for ch in log_entry:
    if ch.isupper():
        uppercase+=1
    elif ch.islower():
        lowercase+=1
    elif ch.isdigit():
        digits+=1
    else:
        special+=1
print(f"Uppercase letters:{uppercase}")
print(f"Lowercase letters:{lowercase}")
print(f"Digits:{digits}")
print(f"Special characters:{special}")
```

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

```
import string
str1=input()
str2=input()
words = str1.split()
cleaned_words = [word.strip(string.punctuation)for word in words]
count = cleaned_words.count(str2)
```

```
print(count)
```

Status : Partially correct

Marks : 7.5/10

4. 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

```
n=int(input())
a=[int(input()) for _ in range(n)]
a.sort()
print("Original List:",a)
b=[]
for i in range(n):
    if i ==0:
        b.append(0)
    elif i%2==0:
        b.append(a[i]**3)
    else:
        b.append(a[i]**2)
print("Replaced List:",b)
```

Status : Correct

Marks : 10/10

5. 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

```
def smallest_missing_positive(nums):  
    nums=set(nums)  
    smallest = 1  
    while smallest in nums:  
        smallest +=1  
    return smallest  
n=int(input())  
arr=list(map(int,input().split()))  
print(smallest_missing_positive(arr))
```

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

```
def find_peaks(measurements):
    peaks=[]
    n=len(measurements)

    for i in range(n):
        if i==0 and n>1 and measurements[i]>measurements[i+1]:
            peaks.append(measurements[i])
        elif i == n-1 and n>1 and measurements[i]>measurements[i-1]:
            peaks.append(measurements[i])
        elif 0<i<n-1 and measurements[i]>measurements[i-1]and measurements[i] >
measurements[i+1]:
            peaks.append(measurements[i])

    return peaks
input_measurements = input()
```

```
measurements_list = list(map(int,input_measurements.split()))  
peak_elements=find_peaks(measurements_list)  
print("Peaks:",peak_elements)
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: Akhil Kumaran
Email: 240701026@rajalakshmi.edu.in
Roll no: 240701026
Phone: 8838850965
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

Gina is working on a data analysis task where she needs to extract sublists from a given list of integers and find the median of each sublist. For each median found, she also needs to determine its negative index in the original list.

Help Gina by writing a program that performs these tasks.

Note: The median is the middle value in the sorted list of numbers, or the first value of the two middle values if the list has an even number of elements.

Example

Input

10

1 2 3 4 5 7 8 9 10 11

3

1 5

2 6

3 10

Output

3 : -8

4 : -7

7 : -5

Explanation

For the first range (1 to 5), the sublist is [1, 2, 3, 4, 5]. The median is 3, and its negative index in the original list is -8.

For the second range (2 to 6), the sublist is [2, 3, 4, 5, 7]. The median is 4, and its negative index in the original list is -7.

For the third range (3 to 10), the sublist is [3, 4, 5, 7, 8, 9, 10, 11]. The median is 7, and its negative index in the original list is -5.

Input Format

The first line of input consists of an integer N, representing the number of elements in the list.

The second line consists of N space-separated integers representing the elements of the list.

The third line consists of an integer R, representing the number of ranges.

The next R lines each consist of two integers separated by space representing the start and end indices (1-based) of the ranges.

Output Format

The output consists of n lines, displaying "X : Y" where X is the median of the

sublist and Y is the negative index in the original list.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10

1 2 3 4 5 7 8 9 10 11

3

1 5

2 6

3 10

Output: 3 : -8

4 : -7

7 : -5

Answer

You are using Python

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

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

```
for _ in range(int(input())):
```

```
    l,r=map(int,input().split())
```

```
    s=sorted(a[l-1:r])
```

```
    m=s[(len(s)-1)//2]
```

```
    print(f"{m} : {a.index(m)-n}")
```

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()
str2=input()
result = ".join(a+b for a,b in zip(str1,str2))
print(result)
```

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=input().strip()
last =input().strip()
role = input().strip().lower()
company = input().strip().lower()

username =first[0].lower() + last.lower()

if role == "admin":
```

```
    email = f"{username}@admin.{company}.com"  
else:  
    email = f"{username}@{company}.com"  
print(email)
```

Status : Correct

Marks : 10/10