Rajalakshmi Engineering College

Name: Kiruthick SM
Email: 240701263@rajalakshmi.edu.in
Roll no: 240701263
Phone: 8122868914
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

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

```
# Function to extract and print the area code from the phone number
def extract_area_code(phone_number):
    # Extract the area code, which is the substring from index 1 to 4 (excluding the parentheses)
    area_code = phone_number[1:4]
    # Output the area code in the required format
    print(f"Area code: {area_code}")

# Input: Phone number in the format (XXX) XXX-XXXX
phone_number = input()

# Call the function with the input phone number
extract_area_code(phone_number)
```

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

```
def rearrange_numbers(lst):
    negative_numbers = [num for num in lst if num < 0]
    positive_numbers = [num for num in lst if num >= 0]
    arranged_list = negative_numbers + positive_numbers
    print("List =", arranged_list)


input_list = eval(input())
```

rearrange_numbers(input_list)

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

```
def manipulate_strings():
    str1 = input()
    str2 = input()
    reversed_str2 = str2[::-1]
    result = str1 + reversed_str2

    print(result)
```

manipulate_strings()

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

```
def slice_string():
    input_string = input()
    start = int(input())
    end = int(input())
    if 0 <= start <= end < len(input_string):
        sliced_string = input_string[start:end+1]
        print(sliced_string)
    else:
        print("Invalid start and end positions")
slice_string()
```

Status : Correct ***Marks : 10/10***

5. 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

```
def manage_list():
    n = int(input())
    lst = []
    for _ in range(n):
        element = int(input())
        lst.append(element)
    M = int(input())
    print(f"List after appending elements: {lst}")
    popped_element = lst.pop(M)
    print(f"List after popping last element: {lst}")
    print(f"Popped element: {popped_element}")

manage_list()
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Kiruthick SM
Email: 240701263@rajalakshmi.edu.in
Roll no: 240701263
Phone: 8122868914
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_PAH

Attempt : 1
Total Mark : 60
Marks Obtained : 60

Section 1 : Coding

1. 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

```
def count_word_occurrences():
    str1 = input()
    str2 = input()
    count = str1.count(str2)
    print(count)

count_word_occurrences()
```

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

```
def transform_list():
    n = int(input())
    lst = [int(input()) for _ in range(n)]

    original_lst = sorted(lst)

    replaced_lst = []

    for i, val in enumerate(original_lst):
        if i == 0:
            replaced_lst.append(0)
```

```
        elif i % 2 == 0:
            replaced_lst.append(val ** 3)
        else:
            replaced_lst.append(val ** 2)

    print(f"Original List: {original_lst}")
    print(f"Replaced List: {replaced_lst}")

transform_list()
```

Status : Correct ***Marks : 10/10***

3. 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

```
def string_operations():
    A = input()
    B = A

    print(f"Length of the string: {len(A)}")
    print(f"Copied string: {B}")

string_operations()
```

Status : Correct ***Marks : 10/10***

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

    peaks = [
        measurements[i]
        for i in range(len(measurements))
        if (i == 0 and measurements[i] > measurements[i+1]) or
           (i == len(measurements) - 1 and measurements[i] > measurements[i-1]) or
           (0 < i < len(measurements) - 1 and measurements[i] > measurements[i-1]
and measurements[i] > measurements[i+1])
    ]

    print(f"Peaks: {peaks}")

find_peaks()
```

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

```
def analyze_log_entry():
    log_entry = input()
```

```
    uppercase_count = sum(1 for char in log_entry if char.isupper())
    lowercase_count = sum(1 for char in log_entry if char.islower())
    digits_count = sum(1 for char in log_entry if char.isdigit())
    special_count = sum(1 for char in log_entry if not char.isalnum())

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

analyze_log_entry()
```

Status : Correct ***Marks : 10/10***

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

    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:
            print(i + 1)
            return

    print(n + 1)

find_missing_positive()
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Kiruthick SM
Email: 240701263@rajalakshmi.edu.in
Roll no: 240701263
Phone: 8122868914
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 : 21

Section 1 : MCQ

1. 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***

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

Answer

25

Status : Correct ***Marks : 1/1***

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

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

Answer

5

Status : Wrong ***Marks : 0/1***

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

Answer

Adds a new element to the end of the list

Status : Correct ***Marks : 1/1***

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

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

Answer

88

Status : Wrong ***Marks : 0/1***

6. 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**

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

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

Answer

nohtyP

Status : Correct **Marks : 1/1**

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

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

Answer

oje

Status : Correct **Marks : 1/1**

9. 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**

10. 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***

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

Answer

list()

Status : Correct ***Marks : 1/1***

12. 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***

13. 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***

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

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

Answer

Python

Status : Correct ***Marks : 1/1***

15. What is the output of the following code?

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

Answer

4

Status : Wrong ***Marks : 0/1***

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 is the output of the following Python code?

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

Answer

HelloWorld

Status : Correct ***Marks : 1/1***

18. 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***

19. 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***

20. 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***

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

Answer

[25, 14, 222, 33, 2]

Status : Wrong ***Marks : 0/1***

22. 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***

23. 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***

24. 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***

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

Answer

extend()

Status : Correct ***Marks : 1/1***

Rajalakshmi Engineering College

Name: Kiruthick SM
Email: 240701263@rajalakshmi.edu.in
Roll no: 240701263
Phone: 8122868914
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_CY

Attempt : 1
Total Mark : 40
Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Nina is working on a project involving multiple sensors. Each sensor provides a data point that needs to be processed to compute an aggregated value.

Given data points from three sensors, write a program to calculate the aggregated value using specific bitwise operations and arithmetic manipulations. The final result should be the aggregated value modulo 1000.

Example:

Input:

1 //sensor 1 data

2 //sensor 2 data

3 //sensor 3 data

Output

9

Explanation

Calculate the bitwise AND of sensor 1 data and sensor 2 data: 0

Calculate the XOR of the result from step 1 and sensor 3 data: 3

Multiply the result from step 2 by 3: 9

Compute the final aggregated value by taking the result from step 3 modulo 1000: 9

So, the aggregated value is 9.

Input Format

The first line of input consists of an integer S1, representing sensor1 data.

The second line of input consists of an integer S2, representing sensor2 data.

The third line of input consists of an integer S3, representing sensor3 data.

Output Format

The output displays an integer representing the aggregated value.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1
2
3
Output: 9

Answer

```
# You are using Python
a=int(input())
b=int(input())
c=int(input())
d=(((a&b)^c)*3)%1000
print(d)
```

Status : Correct **Marks : 10/10**

2. Problem Statement

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

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

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

Input Format

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

Output Format

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

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 10

Output: Number of tacos needed: 25
Total cost: 625

Answer

```
# You are using Python
a=int(input())
if a>=10:
  c=((a*2)+5)
  b=c*25
  print(f"Number of tacos needed: {c}\nTotal cost: {b}")
else:
  c=((a*2))
  if c<20:
    c=20

  b=c*25
  print(f"Number of tacos needed: {c}\nTotal cost: {b}")
```

Status : Correct ***Marks : 10/10***

3. Problem Statement

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

Your task is to guide Olivia through the program.

Input Format

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

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

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

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

Output Format

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

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

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

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

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Alex
10000
350.5
1.5

Output: User Name: Alex
Total Steps: 10000
Calories Burned: 350.5
Workout Duration: 1.5 hours

Answer

```
a=str(input())
b=int(input())
c=float(input())
d=float(input())
print(f"User Name: {a}")
print(f"Total steps: {b}")
print(f"Calories Burned: {c}")
print(f"Workout Duration: {d} hours")
```

Status : Correct ***Marks : 10/10***

4. Problem Statement

Alex is an air traffic controller who needs to record and manage flight delays efficiently. Given a flight number, the delay in minutes (as a string), and the coordinates of the flight's current position (as a complex number),

Help Alex convert and store this information in a structured format.

Input Format

The first line of input consists of an integer N, representing the flight number.

The second line consists of a string representing the delay in minutes.

The third line consists of two floats separated by a space, representing the real and imaginary parts of the complex number for the flight's position.

Output Format

The first line of output displays the complex number.

The second line displays a string with the flight number, delay, and the real and imaginary parts of the complex number, separated by commas.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 12345
30.5
12.3 45.6

Output: (12.3+45.6j)
12345, 30.5, 12.3, 45.6

Answer

```
a=int(input())
b=input().strip()
real,imaginery=map(float,input().split())
c=complex(real,imaginery)
print(c)
print(f"{a},{b},{real},{imaginery}")
```

Status : Correct ***Marks : 10/10***