Rajalakshmi Engineering College

Name: Nishal P
Email: 241501130@rajalakshmi.edu.in
Roll no: 241501130
Phone: 9790965308
Branch: REC
Department: I AI & ML FB
Batch: 2028
Degree: B.E - AI & ML

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

```
n=int(input())
lis=[]
for i in range(n):
   b=int(input())
   lis.append(b)
print("List after appending elements: ",lis)
p=int(input())
l=lis[p]
lis.pop(p)
print("List after popping last element: ",lis)
print("Popped element: ",l)
```

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

```
def extract_area(phone_number):
   area_code=phone_number[1:4]
   return area_code

input_phone=input()
output=extract_area(input_phone)
```

```
print("Area code:",output)
```

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

```
a=input()
b=int(input())
c=int(input())
if b<0 or c>=len(a) or b>c:
   print("Invalid start and end positions")
else:
   print(a[b:c+1])
```

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

4. 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 arrange(input_list):
   neg=[num for num in input_list if num < 0]
   pos=[num for num in input_list if num >= 0]
   arranged_list=neg+pos
   return arranged_list

input_string=input()
input_list=eval(input_string)
output_list=arrange(input_list)
print("List =",output_list)
```

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

5. Problem Statement

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

Ram needs your help to design a program.

Input Format

The input consists of two strings in separate lines.

Output Format

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

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: hello
word

Output: hellodrow

Answer

```
a=input()
b=input()
d=b[::-1]
c=a+d
print(c)
```

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

Rajalakshmi Engineering College

Name: Nishal P
Email: 241501130@rajalakshmi.edu.in
Roll no: 241501130
Phone: 9790965308
Branch: REC
Department: I AI & ML FB
Batch: 2028
Degree: B.E - AI & ML

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

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

```
def find_longest(input_string):
   words=input_string.split()
   longest_word=""
   for word in words:
      if len(word)>len(longest_word):
         longest_word=word

   return longest_word

input_string=input()
longest_word=find_longest(input_string)
print(longest_word)
```

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

2. 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

```
def manage_shopping_list():

    initial_list = input().strip().split()

    # Display the initial list
    print(f"List1: [{', '.join(initial_list)}]")

    # Input the item to be removed
    item_to_remove = input().strip()

    # Remove the item if it exists
    if item_to_remove in initial_list:
        initial_list.remove(item_to_remove)
        print(f"List after removal: [{', '.join(initial_list)}]")
    else:
        print("Element not found in the list")

    # Input the new items to be added
    new_items = input().strip().split()

    # Update the list with new items
    initial_list.extend(new_items)

    # Display the final list
    print(f"Final list: [{', '.join(initial_list)}]")

# Run the shopping list management program
manage_shopping_list()
```

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

3. 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

```
def is_palindrome(num):

    return str(num) == str(num)[::-1]

def process_message(message):

    digit_sum = sum(int(char) for char in message if char.isdigit())
    output = f"{digit_sum}"


    if digit_sum > 9:
        if is_palindrome(digit_sum):
            output += " Palindrome"
        else:
            output += " Not palindrome"

    print(output)


if __name__ == "__main__":

    message = input()
    process_message(message)
```

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