Dashboard / My courses / CD19411-PPD-2022 / WEEK_07-Functions / WEEK-07_CODING

Started on	Monday, 22 April 2024, 8:52 AM
State	Finished
Completed on	Wednesday, 1 May 2024, 12:42 PM
Time taken	9 days 3 hours
Marks	3.00/5.00
Grade	**30.00** out of 50.00 (**60**%)
Name	THEJESH N S 2022-CSD-A

Question **1**

Incorrect

Mark 0.00 out of 1.00

Write a program that reads values from the user until a blank line is entered. Display the total of all of the values entered by the user (or 0 if the first value entered is a blank line). Complete this task using recursion. Your program may not use any loops.
Hint: The body of your recursive function will need to read one value from the
user, and then determine whether or not to make a recursive call. Your function
does not need to take any arguments, but it will need to return a numeric result.

Sample Input

5
10
15
20
25

Sample Output

75

Answer: (penalty regime: 0 %)

Reset answer

```
 1 def readAndTotal():
 2     line=input()
 3        # Read a value from the user
 4 
 5        # Base case: The user entered a blank line so the total is 0
 6        if line ==" ":
 7            return 0
 8 
 9        else:
10            return int(line)+readAndTotal()
11               # Recursive case: Convert the current line to a number and use recursion to read the
12               # subsequent lines
13 
14 # Read the values from the user and compute the total
15 total=readAndTotal()
16 # Display the total
17 print(total)
18 
```

Syntax Error(s)

```
Sorry: IndentationError: unexpected indent (__tester__.python3, line 6)
```

Incorrect

Marks for this submission: 0.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

A list rotation consists of taking the last element and moving it to the front. For instance, if we rotate the list [1,2,3,4,5], we get [5,1,2,3,4]. If we rotate it again, we get [4,5,1,2,3].

Write a Python function rotatelist(l,k) that takes a list l and a positive integer k and returns the list l after k rotations. If k is not positive, your function should return l unchanged. Note that your function should not change l itself, and should return the rotated list.

Here are some examples to show how your function should work.

```
>>> rotatelist([1,2,3,4,5],1)
[5, 1, 2, 3, 4]

>>> rotatelist([1,2,3,4,5],3)
[3, 4, 5, 1, 2]

>>> rotatelist([1,2,3,4,5],12)
[4, 5, 1, 2, 3]
```

Answer: (penalty regime: 0 %)

Reset answer

```
def rotatelist(l, k):
    # If k is not positive, return the list unchanged
    if k <= 0:
        return l

    # Perform k rotations
    rotated_list = l[-k % len(l):] + l[:-k % len(l)]
    return rotated_list

# Test the function
original_list = [1, 2, 3, 4, 5]
rotations = 2
rotated_result = rotatelist(original_list, rotations)

```

	Test	Expected	Got	
✓	`print(rotatelist([1,2,3,4,5],1))`	`[5, 1, 2, 3, 4]`	`[5, 1, 2, 3, 4]`	✓
✓	`print(rotatelist([1,2,3,4,5],3))`	`[3, 4, 5, 1, 2]`	`[3, 4, 5, 1, 2]`	✓
✓	`print(rotatelist([1,2,3,4,5],12))`	`[4, 5, 1, 2, 3]`	`[4, 5, 1, 2, 3]`	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Given an integer n, return an list of length n + 1 such that for each i (0 <= i <= n), ans[i] is the number of 1's in the binary representation of i.

Example:

```
Input: n = 2
Output: [0,1,1]
Explanation:
0 --> 0
1 --> 1
2 --> 10
```

Example2:

```
Input: n = 5
Output: [0,1,1,2,1,2]
Explanation:
0 --> 0
1 --> 1
2 --> 10
3 --> 11
4 --> 100
5 --> 101
```

Note: Complete the given function alone

For example:

Test	Result
print(CountingBits(5))	[0, 1, 1, 2, 1, 2]

Answer: (penalty regime: 0 %)

Reset answer

```
def CountingBits(n):
    result = []
    for i in range(n + 1):
        count = 0
        num = i
        while num > 0:
            count += num & 1
            num >>= 1
        result.append(count)
    return result

```

	Test	Expected	Got	
✓	print(CountingBits(2))	[0, 1, 1]	[0, 1, 1]	✓

	Test	Expected	Got	
✓	`print(CountingBits(5))`	`[0, 1, 1, 2, 1, 2]`	`[0, 1, 1, 2, 1, 2]`	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **4**

Correct

Mark 1.00 out of 1.00

The notion of a palindrome was introduced previously. In this exercise you will write a recursive function that determines whether or not a string is a palindrome. The empty string is a palindrome, as is any string containing only one character. Any longer string is a palindrome if its first and last characters match, and if the string formed by removing the first and last characters is also a palindrome.

Write a program that reads a string from the user and uses your recursive
function to determine whether or not it is a palindrome. Then your program should
display an appropriate message for the user.

Sample Input

malayalam

Sample Output

That was a palindrome!

Sample Input

madan

Sample Output

That is not a palindrome.

Answer: (penalty regime: 0 %)

Reset answer

```
def isPalindrome(s):
    # Base case: The empty string is a palindrome. So is a string containing only 1 character.
    if len(s) <= 1:
        return "That was a palindrome!"

    # Recursive case: The string is a palindrome only if the first and last characters match, and
    # the rest of the string is a palindrome
    if s[0] == s[-1]:
        return isPalindrome(s[1:-1])
    else:
        return "That is not a palindrome."

# Read the string from the user
line = input()

# Check its status and display the result
result = isPalindrome(line)
print(result)

```

	Input	Expected	Got	
✓	malayalam	That was a palindrome!	That was a palindrome!	✓
✓	madan	That is not a palindrome.	That is not a palindrome.	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **5**

Incorrect

Mark 0.00 out of 1.00

In this exercise you will write a function that determines whether or not a password is good. We will define a good password to be a one that is at least 8 characters long and contains at least one uppercase letter, at least one lowercase letter, and at least one number. Your function should return True if the password passed to it as its only parameter is good. Otherwise it should return False. Include a main program that reads a password from the user and reports whether or not it is good. Ensure that your main program only runs when your solution has not been imported into another file.

Sample Input 1

chennai

Sample Output 1

That isn't a good password.

Sample Input 2

Chennai18

Sample Output 2

That's a good password.

Answer: (penalty regime: 0 %)

Reset answer

```
def checkPassword(password):
    # Check if the password is at least 8 characters long
    if len(password) < 8:
        return False

    # Initialize flags for uppercase, lowercase, and digit
    has_upper = False
    has_lower = False
    has_digit = False

    # Check each character in the password
    for char in password:
        if char.isupper():
            has_upper = True
        elif char.islower():
            has_lower = True
        elif char.isdigit():
            has_digit = True

        # Break early if all conditions are met
        if has_upper and has_lower and has_digit:
            break
```

	Test	Expected	Got	
✓	`checkPassword('chennai')`	`That isn't a good password.`	`That isn't a good password.`	✓
✗	`checkPassword('Chennai18')`	`That's a good password.`		✗

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Show differences

Incorrect

Marks for this submission: 0.00/1.00.

◄ Week-07_MCQ

Jump to...

WEEK-07-Extra ►