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

Started on	Friday, 10 May 2024, 12:06 PM
State	Finished
Completed on	Wednesday, 15 May 2024, 7:13 PM
Time taken	5 days 7 hours
Marks	5.00/5.00
Grade	**50.00** out of 50.00 (**100**%)
Name	JEESHAN R J 2022-CSD-A

Question **1**

Correct

Mark 1.00 out of 1.00

Euclid was a Greek mathematician who lived approximately 2,300 years ago. His algorithm for computing the greatest common divisor of two positive integers, a and b, is both efficient and recursive. It is outlined below:

```
If b is 0 then
   eturn a
Else
   Set c equal to the remainder when a is divided by b
   Return the greatest common divisor of b and c
```

Write a program that implements Euclid's algorithm and uses it to determine the greatest common divisor of two integers entered by the user. Test your program with some very large integers. The result will be computed quickly, even for huge numbers consisting of hundreds of digits, because Euclid's algorithm is extremely efficient.

Answer: (penalty regime: 0 %)

```python
def gcd(a,b):
    if a==0:
        return b
    else:
        return gcd(b%a,a)
a=int(input())
b=int(input())
print(gcd(a,b))
```

	Input	Expected	Got	
✓	8 12	4	4	✓
✓	720 1000	40	40	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 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(input1):
    c=0
    c1=0
    c2=0
    for i in input1:
        if i.isupper():
            c+=1
        elif i.islower():
            c1+=1
        else:
            c2+=1
    if c>=1 and c1>=1 and c2>=1:
        print("That's a good password.")
    else:
        print("That isn't a good password.")


```

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

Question **3**

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<=0:
    return I
  else:
   k%= len(l)
   return l[-k:]+l[:-k]
```

	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 **4**

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

```python
def CountingBits(n):
  count=[]
  i=0
  for num in range(0,n+1):
    s=0
    while num!=0:
      if num%2==1:
        s+=1
      num=num//2
    count.append(s)
  return count

```

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

Correct

Mark 1.00 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

```
def readAndTotal():
    value = input()
    if value == '':
        return 0
    else:
        return int(value) + readAndTotal()

total = readAndTotal()
print( total)
```

	Input	Expected	Got	
✓	5 10 15 20 25	75	75	✓

Passed all tests! ✓

Correct