

Started on	Monday, 2 November 2020, 6:24 PM
State	Finished
Completed on	Monday, 2 November 2020, 6:57 PM
Time taken	32 mins 57 secs
Grade	7.00 out of 7.00 (100%)

Question **1**

Correct

Mark 1.00 out of 1.00

Defina o predicado `noIntervalo()` que recebe um inteiro `n` mais dois inteiros `a,b` e verifica se $(n \in [a,b])$

For example:

Test	Result
# 4 pertence a [0,10] <code>print(noIntervalo(4,0,10))</code>	True

Answer: (penalty regime: 0 %)

```
1 def noIntervalo(n, a, b):  
2     return a <= n and n <= b  
3
```

	Test	Expected	Got	
✓	# 4 pertence a [0,10] <code>print(noIntervalo(4,0,10))</code>	True	True	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2

Correct

Mark 1.00 out of 1.00

Sejam quatro valores positivos (a,b,c,d) . Se considerarmos estes valores como comprimentos de segmentos de reta, será que estes segmentos formam um rectângulo?

Defina um predicado que resolva este problema.

For example:

Test	Result
<code>print(eRetangulo(1,1,2,2))</code>	True

Answer: (penalty regime: 0 %)

Reset answer

```
1 def eRetangulo(a, b, c, d):  
2     return (a == b and c == d) or (a == c and b == d) or (a == d and b == c)  
3
```

	Test	Expected	Got	
✓	<code>print(eRetangulo(1,1,2,2))</code>	True	True	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Defina a função `somaKPotencias()` que dado um número k e uma lista de números $[x_0, \dots, x_n]$ calcule

$$\sum_{i=0}^n k \times x_i^k$$
For example:

Test	Result
<code>print(somaKPotencias(3, [1,2,3,4,5]))</code>	675

Answer: (penalty regime: 0 %)

Reset answer

```
1 def somaKPotencias(k, xs, idx = 0):  
2     return sum([k * x**k for x in xs])
```

	Test	Expected	Got	
✓	<code>print(somaKPotencias(3, [1,2,3,4,5]))</code>	675	675	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4

Correct

Mark 2.00 out of 2.00

Um **pangrama** é uma frase que contém todas as letras do alfabeto (vamos considerar o alfabeto de 26 letras, com k,w,y).

Defina o predicado **ePangrama()** que recebe uma **string** e valida se esta é um pangrama.

nota: para simplificar a função será testada apenas com frases com letras minúsculas.

For example:

Test	Result
print(ePangrama('the quick brown fox jumps over the lazy dog'))	True

Answer: (penalty regime: 0 %)

```
1 def ePangrama(txt):
2     txt = txt.lower()
3     aASCII = ord('a')
4     for letra in range(0, 26):
5         if chr(aASCII + letra) not in txt:
6             return False
7     return True
8
```

	Test	Expected	Got	
✓	print(ePangrama('the quick brown fox jumps over the lazy dog'))	True	True	✓

Passed all tests! ✓

Correct

Marks for this submission: 2.00/2.00.

Question 5

Correct

Mark 2.00 out of 2.00

No triângulo de Pascal, cada linha é calculada pela soma da linha anterior: cada número é a soma dos números imediatamente acima.

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
```

Defina a função `linhaPascal()` que recebe uma lista com uma dada linha do triângulo, e calcula a linha seguinte.

For example:

Test	Result
<code>print(linhaPascal([1,3,3,1]))</code>	<code>[1, 4, 6, 4, 1]</code>

Answer: (penalty regime: 0 %)

```
1 def meio(lst):
2     return len(lst) // 2
3
4 def linhaPascal(linha):
5     lst = [1, 1]
6
7     for x in range(meio(linha)):
8         prox = linha[x] + linha[x+1]
9         meioLst = meio(lst)
10
11         lst.insert(meioLst, prox)
12     if x != meio(linha) - 1:
13         lst.insert(meioLst, prox)
14     return lst
```

	Test	Expected	Got	
✓	<code>print(linhaPascal([1,3,3,1]))</code>	<code>[1, 4, 6, 4, 1]</code>	<code>[1, 4, 6, 4, 1]</code>	✓

Passed all tests! ✓

Correct

Marks for this submission: 2.00/2.00.



PREVIOUS ACTIVITY
Capítulo 5 - Iteração

NEXT ACTIVITY
Capítulo 6 - Boas Práticas

