

| | |
|---------------------|-------------------------------------|
| Started on | Saturday, 14 November 2020, 2:46 PM |
| State | Finished |
| Completed on | Saturday, 14 November 2020, 2:51 PM |
| Time taken | 5 mins 23 secs |
| Grade | 9.00 out of 9.00 (100%) |

Question **1**

Correct

Mark 1.00 out of 1.00

Defina a função `ordenarTriplo` que recebe três números a, b, c e devolve um triplo com estes números por ordem decrescente. Por exemplo, `ordenarTriplo(6,1,3)` devolve `(6,3,1)`.

For example:

| Test | Result |
|--|------------------------|
| <code>print(ordenarTriplo(3,2,1))</code> | <code>(3, 2, 1)</code> |

Answer: (penalty regime: 0 %)

```
1 def ordenarTriplo(a, b, c):
2     lista = []
3     if a > b:
4         lista = [a, b]
5     else:
6         lista = [b, a]
7
8     for i in range(2):
9         if c >= lista[i]:
10            lista.insert(i, c)
11            break
12
13     if len(lista) != 3:
14         lista.append(c)
15     return tuple(lista)
```

| | Test | Expected | Got | |
|---|--|------------------------|------------------------|---|
| ✓ | <code>print(ordenarTriplo(3,2,1))</code> | <code>(3, 2, 1)</code> | <code>(3, 2, 1)</code> | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

Defina a função **mediana** que recebe uma lista de inteiros ordenada e devolve a mediana. Se a lista tiver um tamanho ímpar, a mediana será o valor do meio. Se o tamanho for par, a função deve devolver a média dos dois valores do meio. A função deve devolver um resultado do tipo *float*.

For example:

| Test | Result |
|--|--------|
| <code>print(mediana([1,2,3,4,5,6]))</code> | 3.5 |
| <code>print(mediana([1,2,3,4,5]))</code> | 3.0 |

Answer: (penalty regime: 0 %)

```
1 def mediana(lista):
2     comp = len(lista)
3     if comp % 2 == 0:
4         return float((lista[comp // 2 - 1] + lista[comp // 2]) / 2)
5     else:
6         return float(lista[comp // 2])
```

| | Test | Expected | Got | |
|---|--|----------|-----|---|
| ✓ | <code>print(mediana([1,2,3,4,5,6]))</code> | 3.5 | 3.5 | ✓ |
| ✓ | <code>print(mediana([1,2,3,4,5]))</code> | 3.0 | 3.0 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 2.00 out of 2.00

Defina a função recursiva `numDigitos` que recebe um número inteiro positivo e calcula o número de dígitos desse número.

For example:

| Test | Result |
|--------------------------------------|--------|
| <code>print(numDigitos(1234))</code> | 4 |

Answer: (penalty regime: 0 %)

```
1 def numDigitos(n):  
2     if n == 0:  
3         return 0  
4     return 1 + numDigitos(n // 10)
```

| | Test | Expected | Got | |
|---|--------------------------------------|----------|-----|---|
| ✓ | <code>print(numDigitos(1234))</code> | 4 | 4 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 2.00/2.00.

Question 4

Correct

Mark 2.00 out of 2.00

Defina a função recursiva `minimo` que, dada uma lista de números, devolva o menor número da lista.

For example:

| Test | Result |
|---|--------|
| <code>print(minimo([5,4,3,8,1,7,9,8]))</code> | 1 |

Answer: (penalty regime: 0 %)

```
1 def minimo(xs):  
2     comp = len(xs)  
3     if comp == 0:  
4         return []  
5  
6     cabeca = xs[0]  
7     if comp == 1:  
8         return cabeca  
9  
10    res = minimo(xs[1:])  
11    return cabeca if cabeca<res else res
```

| | Test | Expected | Got | |
|---|---|----------|-----|---|
| ✓ | <code>print(minimo([5,4,3,8,1,7,9,8]))</code> | 1 | 1 | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 2.00/2.00.

Question 5

Correct

Mark 3.00 out of 3.00

Defina a função recursiva `apenas_5_3` que verifica se um número inteiro positivo pode ser obtido pelas seguintes operações:

1. Começar pelo número 3 ou 5
2. Adicionar 5 ou multiplicar por 3 o valor atual
3. Voltar ao passo 2 ou terminar

For example:

| Test | Result |
|--|--------|
| <code>print(apenas_5_3(14)) # (3 * 3) + 5</code> | True |
| <code>print(apenas_5_3(60)) # ((5 * 3) + 5) * 3</code> | True |
| <code>print(apenas_5_3(51))</code> | False |

Answer: (penalty regime: 0 %)

```
1 def apenas_5_3(num):
2     if num in (5, 3):
3         return True
4     if num < 5:
5         return False
6
7     if num % 3 == 0:
8         if apenas_5_3(num / 3):
9             return True
10
11     return apenas_5_3(num - 5)
```

| | Test | Expected | Got | |
|---|--|----------|-------|---|
| ✓ | <code>print(apenas_5_3(14)) # (3 * 3) + 5</code> | True | True | ✓ |
| ✓ | <code>print(apenas_5_3(60)) # ((5 * 3) + 5) * 3</code> | True | True | ✓ |
| ✓ | <code>print(apenas_5_3(51))</code> | False | False | ✓ |

Passed all tests! ✓

Correct

Marks for this submission: 3.00/3.00.



PREVIOUS ACTIVITY
Capítulo 6 - Boas Práticas

NEXT ACTIVITY
Comentário Avaliação Contínua 3

