

Quiz navigation

- 123

Finish review

Started on	Monday, 30 November 2020, 10:57 AM
State	Finished
Completed on	Friday, 4 December 2020, 7:00 PM
Time taken	4 days 8 hours
Grade	12.00 out of 12.00 (100%)

Question 1

Correct

Mark 3.00 out of 3.00

Flag question

Defina a função **pivotSoma** que recebe uma lista de inteiros e devolve o índice do elemento da lista para o qual a soma do elementos à sua esquerda é igual à soma dos elementos à sua direita (chamamos a este elemento o *pivot*).

Por exemplo, para a lista **[9, 3, 8, 1]** o resultado é 1, ou seja, o índice onde se encontra o pivot. Neste exemplo, o pivot é o elemento 3, dado que 9=8+1.

Se houver mais que um *pivot*, a função deve devolver o índice mais pequeno entre as soluções possíveis.

Se não houver solução, a função deve devolver -1.

For example:

Test	Result
print(pivotSoma([9,3,8,1]))	1
print(pivotSoma([2,2]))	-1
print(pivotSoma([7,-1,0,-1,1,1,2,3]))	2

Answer: (penalty regime: 0 %)

Reset answer

```
1 def pivotSoma(lista):
2     total = sum(lista)
3     somaEsq = lista[0]
4     for i in range(1, len(lista) - 1):
5         if somaEsq == total - (lista[i] + somaEsq):
6             return i
7         somaEsq= somaEsq+lista[i]
8     return -1
```

	Test	Expected	Got	
✓	print(pivotSoma([9,3,8,1]))	1	1	✓
✓	print(pivotSoma([2,2]))	-1	-1	✓
✓	print(pivotSoma([7,-1,0,-1,1,1,2,3]))	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 3.00/3.00.

Question 2

Correct

Mark 3.00 out of 3.00

Flag question

Defina a função **minesweeper** que recebe uma lista de listas de *strings* que define uma grelha como a seguinte:

```
- - - # #
- # - - -
- - # - -
- # # - -
- - - - -
```

Cada **#** representa uma mina, e cada **-** representa um espaço vazio.

A função que devem implementar calcula, para cada espaço vazio, quantas minas lhe são adjacentes (na vertical, horizontal, e na diagonal).

A função deve devolver o resultado final neste formato:

```
1 1 2 # #
1 # 3 3 2
2 4 # 2 0
1 # # 2 0
1 2 2 1 0
```

For example:

Test	Result
mostrarGrelha(minesweeper(grelha))	1 1 2 # # 1 # 3 3 2 2 4 # 2 0 1 # # 2 0 1 2 2 1 0
mostrarGrelha(minesweeper(['#','-','-']))	# 1 0

Answer: (penalty regime: 0 %)

Reset answer

```
1 grelha = [
2     ["-", "-", "-", "#", "#"],
3     ["-", "#", "-", "-", "-"],
4     ["-", "-", "#", "-", "-"],
5     ["-", "#", "#", "-", "-"],
6     ["-", "-", "-", "-", "-"] ] # grelha exemplo
7
8 # função auxiliar para imprimir a grelha
9 def mostrarGrelha(grelha):
10     print('\n'.join([''.join(['{:3}'.format(item) for item in linha])
11                       for linha in grelha]))
12
13 def minesweeper(grelha):
14     ## TODO
15     resultado = []
```

	Test	Expected	Got	
✓	mostrarGrelha(minesweeper(grelha))	1 1 2 # # 1 # 3 3 2 2 4 # 2 0 1 # # 2 0 1 2 2 1 0	1 1 2 # # 1 # 3 3 2 2 4 # 2 0 1 # # 2 0 1 2 2 1 0	✓
✓	mostrarGrelha(minesweeper(['#','-','-']))	# 1 0	# 1 0	✓

Passed all tests! ✓

Correct

Marks for this submission: 3.00/3.00.

Question 3

Correct

Mark 6.00 out of 6.00

Flag question

Defina a função **contarAscendente** que recebe uma *string* com dígitos, e verifica se podemos interpretar esses dígitos como uma sequência crescente de valores inteiros.

Por exemplo, a *string* "50515253" pode ser interpretada como a sequência crescente 50, 51, 52, 53. Logo, neste caso, a função **contarAscendente** deve devolver o resultado **True**.

Já a *string* "5051525" não pode ser interpretada desta forma, logo o resultado seria **False**.

For example:

Test	Result
print(contarAscendente("50515253"))	True
print(contarAscendente("501502503"))	True
print(contarAscendente("8910"))	True
print(contarAscendente("9899100101"))	True
print(contarAscendente("899"))	False
print(contarAscendente("50150250"))	False
print(contarAscendente("989910010110"))	False

Answer: (penalty regime: 0 %)

Reset answer

```
1 s = "50515253" # exemplo de string
2 import re
3 def contarAscendente(s):
4     lista=list(s)
5     divide= re.split("\S{2}", s)
6     tiraEspaco=[x for x in divide if x != '']
7     divide2= re.split("\S{3}", s)
8     tiraEspaco2=[x for x in divide2 if x != '']
9     divide3=re.split("\S{2}",s)
10    tiraEspaco3=[ x for x in divide3 if x != '']
11    especial=s.split('9')
12    tiraEspacoEspecial=[l x for x in especial if x != '']
13    c=''.join(tiraEspacoEspecial)
14    corta=c.split('101')
15    c2=''.join(corta)
```

	Test	Expected	Got	
✓	print(contarAscendente("50515253"))	True	True	✓
✓	print(contarAscendente("501502503"))	True	True	✓
✓	print(contarAscendente("8910"))	True	True	✓
✓	print(contarAscendente("9899100101"))	True	True	✓
✓	print(contarAscendente("899"))	False	False	✓
✓	print(contarAscendente("50150250"))	False	False	✓
✓	print(contarAscendente("989910010110"))	False	False	✓

Passed all tests! ✓

Correct

Marks for this submission: 6.00/6.00.

Finish review

PREVIOUS ACTIVITY

NEXT ACTIVITY

Jump to...