

Pergunta **1**

Parcialmente correta

Nota: 3,00

Sistema de planeamento de itinerário - leia o [enunciado completo](#) antes de começar a responder.

Neste trabalho pretende-se implementar funções que permitam planear itinerários que passem através de vários pontos geográficos. Considere que dispõe de um dicionário com informação sobre a localização de várias cidades portuguesas em coordenadas GPS. Este dicionário está disponível num ficheiro cidades.py que pode encontrar no separador [Documentos](#) do Moodle.

Deverá escrever as seguintes funções, recorrendo ao que aprendeu sobre programação funcional e funções de ordem superior.

1. Uma função `distancia_itinerario(itinerario)` que calcula a distância total de um itinerario (em km), ou seja, a soma das distâncias entre duas cidades consecutivas.
2. Uma função `adicionar_cidade(itinerario, cidade)` que adiciona uma cidade a um itinerário já existente. Esta cidade deve ser colocada entre duas cidades do itinerário original, de modo a minimizar o desvio adicional. As cidades inicial e final do itinerário não deverão ser alteradas.
3. Uma função `construir_itinerario(origem, destino, lista_cidades)` que constrói um itinerário a partir de uma lista de cidades, usando a função do passo anterior. A sua implementação deverá adicionar sequencialmente cada cidade da lista dada ao itinerário, de forma a minimizar a distância percorrida. As cidades inicial (origem) e final (destino) do itinerário não deverão ser alteradas.

Para cada uma das três funções assinaladas deverá incluir, para além do docstring, uma bateria de testes, utilizando a técnica da partição do espaço de entrada.

For example:

Test	Result
<pre>print(round(distancia_itinerario(['Lisboa', 'Setúbal', 'Coimbra', 'Aveiro', 'Viseu', 'Porto']), 3))</pre>	419.256
<pre>print(adicionar_cidade(['Lisboa', 'Setúbal', 'Coimbra', 'Viseu', 'Porto'], 'Aveiro'))</pre>	<pre>['Lisboa', 'Setúbal', 'Coimbra', 'Viseu', 'Aveiro', 'Porto']</pre>

Test	Result
<pre>print(construir_itinerario('Lisboa', 'Porto', ['Viseu', 'Coimbra', 'Aveiro', 'Setúbal']))</pre>	<pre>['Lisboa', 'Setúbal', 'Coimbra', 'Viseu', 'Aveiro', 'Porto']</pre>

Answer: (penalty regime: 0 %)

Reset answer

```

1 import itertools
2 import math as m
3 import functools
4
5 ### Dicionário com as cidades - NÃO ALTERAR!
6
7 cidades = {'Lisboa': (38.7452, -9.1604),
8            'Vila Nova de Gaia': (41.1333, -8.6167),
9            'Porto': (41.1495, -8.6108),
10           'Braga': (41.5333, -8.4167),
11           'Matosinhos': (41.2077, -8.6674),
12           'Amadora': (38.75, -9.2333),
13           'Almada': (38.6803, -9.1583),
14           'Oeiras': (38.697, -9.3017).
15

```

Submeter

	Test	Expected	Got	
✓	<pre>print(round(distancia_itinerario(['Lisboa', 'Setúbal', 'Coimbra', 'Aveiro', 'Viseu', 'Porto']), 3))</pre>	419.256	419.256	✓

	Test	Expected	Got	
✖	<code>print(adicionar_cidade(['Lisboa', 'Setúbal', 'Coimbra', 'Viseu', 'Porto'], 'Aveiro'))</code>	<code>['Lisboa', 'Setúbal', 'Coimbra', 'Viseu', 'Aveiro', 'Porto']</code>	<code>['Lisboa', 'Setúbal', 'Coimbra', 'Aveiro', 'Viseu', 'Porto']</code>	✖
✖	<code>print(construir_itinerario('Lisboa', 'Porto', ['Viseu', 'Coimbra', 'Aveiro', 'Setúbal']))</code>	<code>['Lisboa', 'Setúbal', 'Coimbra', 'Viseu', 'Aveiro', 'Porto']</code>	<pre>***Error*** Traceback (most recent call last): File "__tester__.python3", line 299, in <module> print(construir_itinerario('Lisboa', 'Porto', ['Viseu', 'Coimbra', 'Aveiro', 'Setúbal'])) NameError: name 'construir_itinerario' is not defined</pre>	✖

Testing was aborted due to error.

[Show differences](#)



[PREVIOUS ACTIVITY](#)
[Trabalho 2](#)

[NEXT ACTIVITY](#)
[Ficha 1](#)

