



O tempo médio de voo entre dois locais distintos pode ser obtido por meio da seguinte tabela:

	A	B	C	D	E	F	G
A	00:00	02:00	11:00	06:00	15:00	11:00	01:00
B	02:00	00:00	07:00	12:00	04:00	02:00	15:00
C	11:00	07:00	00:00	11:00	08:00	03:00	13:00
D	06:00	12:00	11:00	00:00	10:00	02:00	01:00
E	15:00	04:00	08:00	10:00	00:00	05:00	13:00
F	11:00	02:00	03:00	02:00	05:00	00:00	14:00
G	01:00	15:00	13:00	01:00	13:00	14:00	00:00

Elabore um programa completo para que uma pessoa interessada em viajar de uma local para outro possa estimar o tempo de viagem. Você terá de fazer o seguinte:

1 – Construa uma função que leia um arquivo (**locais.txt**) e crie uma lista contendo todos os locais existentes em uma tabela de tempos de voo. A lista a seguir representa os locais presentes na tabela acima.

["A ", "B", "C", "D", "E", "F", "G "]

O primeiro dado a ser lido do arquivo é a quantidade de locais (QC) que estão presentes na tabela (no exemplo acima, 7). Após isso, QC strings deverão ser lidas e inseridas em uma lista de strings.

O cabeçalho dessa função é o seguinte:

```
def carregaLocais():
```

Ela deve retornar uma lista com os nomes dos locais.



2 – Construa uma função que leia um arquivo (**distancias.txt**) e crie uma lista contendo QC elementos, em que cada elemento é uma sublista contendo QC tempos de voo entre dois locais. O exemplo a seguir exibe a sublista com os tempos de voo entre o local A e os seis demais locais presentes na tabela acima.

```
["00:00", "02:00", "11:00", "06:00", "15:00", "11:00", "01:00"]
```

O primeiro dado a ser lido do arquivo é a quantidade de locais (QC) que estão presentes na tabela (no exemplo acima, 7). O número de elementos (tempos de voo) de cada sublista é QC, pois a tabela acima é sempre uma matriz quadrada. Dessa forma, um arquivo que contenha os tempos de voo presentes na tabela acima terá  $7^2 = 49$  tempos de voo.

Na lista de tempos (lt) que será criada para representar uma tabela de tempos de voo, o elemento `lt[i][j]` representa a distância entre o local `[i]` (lista de locais) e o local `[j]` (lista de locais).

O cabeçalho dessa função é o seguinte:

```
def carregaDistancias():
```

Ela deve retornar a lista de tempos de voo.

3 – Construa uma função que receba uma lista de locais e um local. A função deve retornar o índice do local na lista, caso ele pertença à mesma, ou -1, caso ele não pertença à lista.

O cabeçalho dessa função é o seguinte:

```
def busca(lista, local):
```

4 – Construa um bloco principal que leia continuamente do teclado um par de locais e exiba o tempo estimado de voo entre eles. Caso um local não seja encontrado uma mensagem adequada deve ser exibida.

Leia, primeiramente, o local de origem e realize uma busca na lista. Caso a origem seja encontrada, leia um destino e realize uma nova busca. Trate a possibilidade de o usuário informar um mesmo local para a origem e para o destino. Nesse caso, exiba uma mensagem apontando um erro nos dados informados.

A leitura da string "###" indica o término dos dados.