Aula prática nº 6 – Sequências

Exercícios

1) Tente adivinhar o resultado de cada uma das instruções abaixo. Algumas não têm resultado e outras dão erros. Use o Python em modo interativo para confirmar. (Veja se acerta mais que o/a colega do lado!)

```
lst = [5,3,8,7]
len(lst)
lst[2]
lst[4]
lst[-4]
lst[1:3]
lst[:-1]
lst[2:2]
lst[2:2] = [99] ; lst
lst.append(33) ; lst
```

```
t = ("ana", (1974,4,25))
len(t)
t[1]
t[1][0]
(1974,3,30) < (1974,4,1)
(1974,3) < (1974,3,-1,-2)
s = "abcde"
s[2:] + s[:2]
s > "abel"
```

2) Siga os seguintes passos, testando cada um:
   a) Crie uma função que leia uma sequência de números introduzidos pelo utilizador e os devolva numa lista. O utilizador deve introduzir um número por linha e indicar o fim da lista com uma linha vazia.
   b) Crie uma função `countLower(lst, v)` que conte (e devolva) quantos elementos da lista `lst` são inferiores ao valor `v`.
   c) Crie uma função `minmax(lst)` que devolva o mínimo e o máximo de uma lista de valores. Consegue fazê-la sem usar as funções `min` e `max`?
   d) Recorra às funções anteriores para fazer um programa que leia uma lista de números, determine o valor médio entre o mínimo e o máximo e conte quantos números são inferiores a esse valor.

3) O programa `telephones.py` define duas listas, uma com números de telefone e outra com os nomes correspondentes.

```
telList = ['975318642', '234000111', '777888333', ...]
nameList = ['Angelina', 'Brad',      'Claudia',   ...]
```

a) Complete a função `telToName` que, dado um número de telefone (e as duas listas), devolve o nome respetivo (ou o próprio número, se não estiver na lista). Isto é o que os telemóveis fazem quando recebem uma chamada.

b) Complete a função `nameToTels` que, dada parte de um nome, devolve a lista dos números correspondentes a nomes que incluem essa parte. (Como quando pesquisa na lista de contactos do telemóvel.)

c) Corra o programa para testar essas funções.

4) Escreva uma função que, dada uma lista de equipas de futebol, gere uma lista de todos os jogos que se podem fazer entre elas. Por exemplo:

```
allMatches(["SCP", "SLB", "FCP"])  →
[("SCP", "SLB"), ("SCP", "FCP"), ("SLB", "SCP"), ...]
```

Com 3 equipas deve obter 6 jogos, com 4 equipas deve obter 12 jogos. Confirme e teste com ainda mais quipas.

5) Crie uma função que conte quantos dígitos há numa dada string. Por exemplo: `countDigits("23 mil 456")` deve devolver `5`. *Sugestão: o método `isdigit` verifica se uma string só tem dígitos, e.g., `"1".isdigit() → True`.*

6) Crie uma função que, dado um nome, crie uma versão abreviada, formada apenas pelas letras maiúsculas. Por exemplo:

```
shorten("Universidade de Aveiro") → "UA",
shorten("United Nations Organization") → "UNO".
```

Sugestão: o método `isupper` verifica se uma string só tem maiúsculas, e.g., `"A".isupper() → True`.

7) Crie uma função `ispalindrome(s)` que devolva um valor booleano indicando se a string `s` é um palíndromo ou não.