

Fundamentos de Programação de Computadores

Licenciatura em Sistemas de Informação para Gestão

1º Semestre 2021/2022

Ficha de trabalho 7

Objetivos:

- Desenvolver competências Python usando listas e tuplos

Grupo 1 – Listas

- Uma lista é um **conjunto ordenado de valores**, cada um dos quais **identificado por um índice**.
- A cada um desses valores chama-se **elemento**.
- Semelhante a uma String mas os seus **elementos são mutáveis**, i.e., podem ser alterados e podem ser de **qualquer tipo**.
- Sintaxe das listas em Python:
 - [1,2,3,4]
 - ["pao", "pao", "queijo", "queijo"]
 - [1, "pao", 2, "queijo", ["preco", 200]]

As listas têm uma série de **métodos** que permitem a **manipulação das mesmas**

<u>Método</u>	<u>Sintaxe</u>	<u>Descrição</u>
append	<code>mylist.append(item)</code>	Adiciona o elemento ao final da lista
insert	<code>mylist.insert(pos, item)</code>	Insere um elemento na posição indicada
pop	<code>mylist.pop()</code>	Remove e retorna o último elemento
pop	<code>mylist.pop(pos)</code>	Remove e retorna o elemento da posição indicada
sort	<code>mylist.sort()</code>	Ordena a lista (crescente ou alfabeticamente)
reverse	<code>mylist.reverse()</code>	Ordena a lista na ordem contrária do sort
index	<code>mylist.index(item)</code>	Retorna a primeira posição com o item indicado
count	<code>mylist.count(item)</code>	Retorna o número de ocorrências do item indicado
remove	<code>mylist.remove(item)</code>	Remove a primeira ocorrência do item indicado

Exemplos da aplicação dos métodos para manipulação de listas:

```
>>> mylist = []
>>> mylist.append(5)
>>> mylist.append(27)
>>> mylist.append(3)
>>> mylist.append(12)
>>> mylist
[5, 27, 3, 12]
>>> mylist.insert(1, 12)
>>> mylist
[5, 12, 27, 3, 12]
>>> mylist.count(12)
2
>>> mylist.index(27)
2
```

```
>>> mylist.reverse()
>>> mylist
[12, 3, 27, 12, 5]
>>> mylist.sort()
>>> mylist
[3, 5, 12, 12, 27]
>>> mylist.remove(5)
>>> mylist
[3, 12, 12, 27]
>>> lastitem = mylist.pop()
>>> lastitem
27
>>> mylist
[3, 12, 12]
>>> seconditem = mylist.pop(1)
>>> seconditem
12
>>> mylist
[3, 12]
```

Tendo em conta estas características resolve os seguintes exercícios utilizando o IDLE ou o Pycharm.

1- Considera um programa que começa com a seguinte atribuição:

```
xs = [12, 10, 32, 3, 66, 17, 42, 99, 20]
```

- Escreve um ciclo for que imprime cada um dos números da lista xs numa linha separada.
- Escreve um outro ciclo em que, em cada linha, se imprime o número, o seu quadrado, e a sua raiz quadrada.
- Escreve um ciclo que soma todos os números em xs usando uma variável auxiliar total, e imprime
- numa linha separada cada um dos números da lista é a soma parcial até esse número.

2- A pataca é a moeda oficial de Macau. Ao câmbio atual, uma pataca vale 0.1127 euros. Implementa a função `mop2eur(p)` que, dada uma lista de valores em patacas, retorna o valor da respetiva soma em euros. Por exemplo, o resultado de `mop2eur([15, 73])` é aproximadamente 9.918.

3- Considera que o preço atual da gasolina é 1.63 euros por litro. Implementa uma função `valor(v)` que, dada a lista `v` de litros abastecidos num mês, retorna o valor total mensal despendido. Por exemplo, o resultado de `valor([24.8, 49.1])` é 120.457.

4- Escreve uma função que conte quantas palavras tem uma frase. Sugestão: Utiliza listas. O python inclui a função `split()` que permite dividir uma string numa lista de palavras, tendo por omissão o espaço como separador.

```
Ex: string = "Eu gosto de programação"
    string = string.split()
    ["Eu", "gosto", "de", "programação"]
```

5- Recorda que um número inteiro `d` é divisor de `n` (ou equivalente, `n` é múltiplo de `d`) se e só se o resto da divisão de `n` por `d` for zero.

a) Escreva uma função `divisores(n)` que calcula a lista dos divisores de `n` inferiores a `n`, por ordem crescente. Exemplo: `divisores(12)` dá `[1, 2, 3, 4, 6]`

b) Um número inteiro é perfeito se for igual à soma dos seus divisores. Exemplo: 6 é perfeito porque $6 = 1+2+3$ mas 10 não é porque $10 \neq 1+2+5$. Escreva uma função `perfeito(n)` que testa se `n` é perfeito ou não. O resultado deve ser um valor lógico.

6- Escreva uma função `ocorrencias(txt, c)` que retorna uma lista com os índices das ocorrências dum carácter `c` na cadeia `txt`. Por exemplo:

```
>>> ocorrencias('banana', 'a')
[1, 3, 5]
```

7- Escreva uma função `media_arit(xs)` cujo resultado é a média aritmética de uma lista, isto é, $A = \frac{x_1 + x_2 + \dots + x_n}{n}$

8- No Totoloto sorteiam-se seis números distintos de 1 a 49 (mais um “número da sorte” que vamos ignorar neste exercício). Escreva uma função `acertos(chave, aposta)` para contabilizar os números certos numa aposta; a chave e a aposta são listas de inteiros por ordem arbitrária, mas que não contêm valores repetidos.

Exemplo:

```
>>> acertos([25, 49, 27, 3, 17, 33], [33, 9, 19, 49, 7, 40])
2
```

Joga com os teus amigos de duas formas:

- Inventar uma sequência de 6 números aleatórios de 1 a 49;
- Ou utiliza a seguinte função que te gera automaticamente uma lista com 5 números aleatórios de 1 a 49.

```
import random
```

```
def listaAleatoria():
```

```
    lista=[]
```

```
    for i in range (7):
```

```
        x = random.randint(1, 50)
```

```
        lista.append(x)
```

```
    return lista
```

- Pede aos teus amigos que tentem adivinhar a lista e apostem um almoço 😊
- Usa a tua função `acertos` para verificar quantos números os teus amigos adivinharam.
- `acertos (listaAleatoria(), chave)`

Bom Trabalho.