

## Oitava Lista de Exercícios - Computational Thinking

1. Escreva um programa que cria uma lista de strings e preenche essa lista com 10 valores que serão digitados pelo usuário. Imprima a lista na tela.
2. Escreva uma função que recebe como parâmetro um inteiro positivo  $n$  e retorna uma lista preenchida com  $n$  números inteiros aleatórios entre 1 e 1000.

```
1 import random
2
3 #gerando um numero aleatorio entre 1 e 1000
4 numero = random.randint(1,1001)
```

3. Escreva um algoritmo que pede para o usuário digitar 10 strings uma de cada vez. Depois que o usuário digitar todas elas, seu programa deverá imprimir as strings na ordem inversa de leitura. Por exemplo se as duas últimas strings foram, respectivamente, avestruz e onça; o programa imprime onça e depois avestruz.
4. Escreva um programa que pede para o usuário digitar um inteiro  $n$ . Depois seu algoritmo pede para o usuário digitar uma sequência de  $n$  números reais. Após a entrada dos dados, seu programa deverá imprimir os resultados das seguintes somas:  $v[0] + v[n - 1]$ ,  $v[1] + v[n - 2]$ ,  $v[2] + v[n - 3]$ , ...; até que todos os valores informados tenham participado de alguma soma.
5. Faça uma função em Python que recebe uma lista de números reais e retorna `True` se a lista está ordenada em ordem crescente ou `False` se ela não está.
6. Escreva uma função em Python que recebe um inteiro  $x$  e uma lista de números inteiros ordenada em ordem crescente. Sua função deverá inserir  $x$  na lista de forma que ela continue ordenada em ordem crescente. Neste exercício você deve usar apenas o método `insert` da lista.  

```
1 def insereOrdenado(x, lista):
2     #desenvolva aqui seu algoritmo
3
4 vet = [1, 6, 10, 24, 25, 30, 45]
5 insereOrdenado(20, vet)
6 print(vet)
```
7. Escreva uma função que recebe como parâmetro uma lista de números reais e um real  $x$ . Sua função deverá contar e retornar a quantidade de elementos que são maiores ou iguais a  $x$ .
8. Escreva uma função que recebe como parâmetro uma lista  $a$  de números inteiros e retorna uma outra lista contendo somente os números pares de  $a$ .
9. Escreva uma função que recebe como parâmetro duas listas `listaA` e `listaB` de números reais. Ela deverá retornar uma terceira lista contendo todos os números da `listaA` que também estão na `listaB`.
10. Escreva um método chamado `intercala` que recebe como parâmetro duas listas  $a$  e  $b$  de números reais ordenadas em ordem crescente. Seu método deverá retornar uma terceira lista contendo a união dos elementos de  $a$  e  $b$  ordenados em ordem crescente.

11. Na Copa do Mundo do Brasil os quadrifinalistas foram, em ordem alfabética: Alemanha, Argentina, Bélgica, Brasil, Colômbia, Costa Rica, França e Holanda. Imaginando que não sabemos os resultados e nem os cruzamentos, escreva um algoritmo que gere todos os possíveis campeões e vice-campeões dentre os oito times.
12. Dado uma lista de strings, escreva um algoritmo que conta o número de ocorrências da string na lista. Por exemplo:

```
1  letras = ['a', 'e', 'b', 'a', 'c', 'a', 'b', 'a', 'e']
```

a: 4 vezes

e: 2 vezes

b: 2 vezes

c: 1 vezes

Dica, crie uma função para ajudar a resolver o problema.

13. Considere um corredor com mil portas, numeradas de 1 a 1000, que se encontram todas fechadas. Por esse corredor passarão mil pessoas, que modificarão o estado das portas cujo número seja múltiplo do seu número de passagem: a pessoa com o número 3 modificará o estado (fechará se estiverem abertas ou abrirá se estiverem fechadas) das portas nº 3, 6, 9, 12, ... e a pessoa com o número 7 fará o mesmo às portas 7, 14, 21, etc. Construa um programa que permita saber quantas são e quais são as portas abertas após a passagem da milésima pessoa. Dica: use uma lista onde você armazena valores `True` ou `False` com 1001 posições para representar as portas. Dica, crie uma função para ajudar a resolver o problema.

Boa sorte!