

MC102 - Algoritmos e Programação de Computadores

Turma Z - Segundo Semestre de 2019

```
# True or False?
def maior(a, b):
    print(a > b)

maior(10, 20)
```

```
def soma(a, b, c):
    return a + b + c

c = 5
soma(0, 5)
```

```
def subtrai(a, b):
    c = a - b

subtrai(10, 20)
print(c)
```

```
def soma(a, b) :
    r = a + b
    a = 0
    b = 0
    return r

a = 10
b = 20
c = soma(a, b)
print(a, b, c)
```

```
def misterio(a, b, c):
    if a < b and a < c :
        print(a)
    elif b < c:
        print(b)
    else:
        print(c)
    return

misterio(5, 6, 1)
```

```
def menos(a) :
    return -a

def subtrai(a, b) :
    return a + menos(b)

a = 10
b = 20
r = subtrai(a,b)
print(menos(r))
```

True or False?

```
def maior(a, b):  
    print(a > b)
```

```
maior(10, 20)
```

False

```
def soma(a, b, c):
```

```
    return a + b + c
```

```
c = 5
```

```
soma(0, 5)
```

soma() missing 1 required positional argument: 'c'

```
def subtrai(a, b):
```

```
    c = a - b
```

```
subtrai(10, 20)
```

```
print(c)
```

name 'c' is not defined

```
def soma(a, b) :
```

```
    r = a + b
```

```
    a = 0
```

```
    b = 0
```

```
    return r
```

```
a = 10
```

```
b = 20
```

```
c = soma(a, b)
```

```
print(a, b, c)
```

10 20 30

```
def misterio(a, b, c):
```

```
    if a < b and a < c :
```

```
        print(a)
```

```
    elif b < c:
```

```
        print(b)
```

```
    else:
```

```
        print(c)
```

```
    return
```

```
misterio(5, 6, 1)
```

1

```
def menos(a) :
```

```
    return -a
```

```
def subtrai(a, b) :
```

```
    return a + menos(b)
```

```
a = 10
```

```
b = 20
```

```
r = subtrai(a,b)
```

```
print(menos(r))
```

10

```
lista = [0, 1, 4, 5, 12]
lista.append(3)
print(lista)
```

```
tupla = ("Ibis", 0, "Porto", 3)
tupla[1] = 10
print(tupla)
```

```
lista = [0, 1, 4, 5, 12]
lista[7] = 3
print(lista)
```

```
lista_tuplas = [("A", 1), ("B", 2)]
lista_tuplas[0] = ("C", 3)
print(lista_tuplas)
```

```
lista = [0, 1, 4, 5, 12]
lista.append(3)
print(lista)
```

```
[0, 1, 4, 5, 12, 3]
```

```
tupla = ("Ibis", 0, "Porto", 3)
tupla[1] = 10
print(tupla)
```

```
'tuple' object does not support item assignment
```

```
lista = [0, 1, 4, 5, 12]
lista[7] = 3
print(lista)
```

```
list assignment index out of range
```

```
lista_tuplas = [("A", 1), ("B", 2)]
lista_tuplas[0] = ("C", 3)
print(lista_tuplas)
```

```
[('C', 3), ('B', 2)]
```

```
A = [[6, 3, 2], [7, 2, 0], [2, 1, 0]]
B = [[1, 2, 4, 5], [2, -2, 0, 1], [1, 0, -1, 4]]
```

```
for i in range(len(A)) :
    for j in range(len(A)) :
        A[i][j] = B[i][j] - 1
print(A[0])
```

```
vitorias = {"Ibis":0, "Porto":1, "Verona":2, "Bonito":1}
partida = ("Itapipoca", 1, "Cruzeiro", 0)
if partida[1] > partida[3] :
    vencedor = partida[0]
else:
    vencedor = partida[2]
vitorias[vencedor] = vitorias[vencedor] + 1
print(vitorias[vencedor])
```

```
A = [[6, 3, 2], [7, 2, 0], [2, 1, 0]]
B = [[1, 2, 4, 5], [2, -2, 0, 1], [1, 0, -1, 4]]
```

```
for i in range(len(A)) :
    for j in range(len(A)) :
        A[i][j] = B[i][j] - 1
print(A[0])
```

```
[0, 1, 3]
```

```
vitorias = {"Ibis":0, "Porto":1, "Verona":2, "Bonito":1}
partida = ("Itapipoca", 1, "Cruzeiro", 0)
if partida[1] > partida[3] :
    vencedor = partida[0]
else:
    vencedor = partida[2]
vitorias[vencedor] = vitorias[vencedor] + 1
print(vitorias[vencedor])
```

```
KeyError: 'Itapipoca'
```

A partir desse slide, utilizaremos o material desenvolvido pela professora Sandra Avila e disponível em <http://www.ic.unicamp.br/~sandra/>

Exercícios

Exercícios

- Escreva um programa que leia todas as posições de uma matriz 10x10. O programa deve então exibir o número de posições não nulas na matriz.

Exercícios: Corrida de Kart

- Uma pista de Kart permite 10 voltas para cada um de 6 corredores. Faça um programa que leia os nomes e os tempos (em segundos) de cada volta de cada corredor e guarde as informações em uma matriz. Ao final, o programa deve informar:
 - De quem foi a melhor volta da prova, e em que volta.
 - Classificação final em ordem crescente.
 - Qual foi a volta com a média mais rápida.

Exercícios: Jogo da Velha

- Faça um programa para o Jogo da Velha. Seu programa deve ler uma matriz 3x3 que representa um tabuleiro de jogo da velha.