



Algoritmos e Programação de Computadores

Revisão: Prova 2

Profa. Sandra Avila

Instituto de Computação (IC/Unicamp)

MC102, 30 Junho, 2020

Conteúdo da Prova 2

- Prova 1: Variáveis, Tipos, Comandos condicionais, Comandos repetitivos, Listas
- Tuplas
- Dicionários
- Funções
- Matrizes e listas multidimensionais
- Algoritmos de busca: Sequencial e Binária
- Recursão
- Algoritmos de ordenação: Bubble, Insertion, Selection, Quick (opcional) e Merge

Exercícios

A

B

C

D

1. O que será impresso pelo programa?

```
tupla1 = (10, 3, "mc102", 2.5)
tupla1[0] = 2
tupla1[1] = 8
tupla2 = tupla1[0:2]
print(tupla2)
```

- a. Não irá compilar.
- b. (2, 8)
- c. (2, 8, "mc102")
- d. (10, 3, "mc102")

1. O que será impresso pelo programa?

```
tupla1 = (10, 3, "mc102", 2.5)
tupla1[0] = 2
tupla1[1] = 8
tupla2 = tupla1[0:2]
print(tupla2)
```

a. Não irá compilar.

b. (2, 8)

c. (2, 8, "mc102")

d. (10, 3, "mc102")

2. Qual é o valor impresso ao final da seguinte sequência de comandos?

```
tupla = (3, 1, 2, 5)
a, b, c, d = tupla
print(a+d)
```

- a. Nada é impresso, ocorre um erro de execução.
- b. 4
- c. 3
- d. 8

2. Qual é o valor impresso ao final da seguinte sequência de comandos?

```
tupla = (3, 1, 2, 5)
a, b, c, d = tupla
print(a+d)
```

- a. Nada é impresso, ocorre um erro de execução.
- b. 4
- c. 3
- d. 8**

3. O que será impresso pelo programa?

```
tupla1 = (3, 1, 2, 5)
tupla2 = tupla1[1:4]
tupla3 = tupla1 + tupla2
print(tupla3)
```

- a. (6, 2, 4, 10)
- b. (3, 1, 2, 5)
- c. (3, 1, 2, 5, 1, 2, 5)
- d. (1, 2, 5)

3. O que será impresso pelo programa?

```
tupla1 = (3, 1, 2, 5)
tupla2 = tupla1[1:4]
tupla3 = tupla1 + tupla2
print(tupla3)
```

- a. (6, 2, 4, 10)
- b. (3, 1, 2, 5)
- c. (3, 1, 2, 5, 1, 2, 5)**
- d. (1, 2, 5)

4. O que imprime o seguinte comando?

```
animais = {"gato":12, "cachorro":6, "elefante":23}  
animais["rato"] = animais["gato"] + animais["cachorro"]  
print(animais["rato"])
```

- a. Não irá compilar.
- b. 12
- c. 6
- d. 18

4. O que imprime o seguinte comando?

```
animais = {"gato":12, "cachorro":6, "elefante":23}  
animais["rato"] = animais["gato"] + animais["cachorro"]  
print(animais["rato"])
```

- a. Não irá compilar.
- b. 12
- c. 6
- d. 18**

5. O que será impresso pelo programa a seguir?

```
animais = {"gato":12,"cachorro":6,"rato":18,"elefante":23}
for i in ["cachorro","rato","elefante","gato"]:
    if i == "gato" or i == "rato":
        print("a",end=" ")
    else:
        print(animais[i],end=" ")
```

- a. a a a a
- b. a 6 a 18
- c. 6 a 23 a
- d. 6 18 23 12

5. O que será impresso pelo programa a seguir?

```
animais = {"gato":12,"cachorro":6,"rato":18,"elefante":23}
for i in ["cachorro","rato","elefante","gato"]:
    if i == "gato" or i == "rato":
        print("a",end=" ")
    else:
        print(animais[i],end=" ")
```

a. a a a a

b. a 6 a 18

c. 6 a 23 a

d. 6 18 23 12

6. Qual dos seguintes é um cabeçalho válido de uma função (a primeira linha da definição da função)?

a. `def desenhaCirculo(t):`

b. `def desenhaCirculo:`

c. `desenhaCirculo(t, tam):`

d. `def desenhaCirculo(t, tam)`

6. Qual dos seguintes é um cabeçalho válido de uma função (a primeira linha da definição da função)?

a. `def desenhaCirculo(t) :`

b. `def desenhaCirculo:`

c. `desenhaCirculo(t, tam) :`

d. `def desenhaCirculo(t, tam)`

7. Quais são os parâmetros da seguinte função?

```
def potencia (base, expoente) :  
    resultado = 1  
    for numero in range (1, expoente+1) :  
        # base ** expoente = base * base (expoente vezes)  
        resultado = resultado * base  
    return resultado
```

- a. resultado
- b. numero, resultado
- c. base, expoente
- d. numero, resultado, base, expoente

7. Quais são os parâmetros da seguinte função?

```
def potencia (base, expoente) :  
    resultado = 1  
    for numero in range (1, expoente+1) :  
        # base ** expoente = base * base (expoente vezes)  
        resultado = resultado * base  
    return resultado
```

- a. resultado
- b. numero, resultado
- c. **base, expoente**
- d. numero, resultado, base, expoente

8. O que o seguinte código imprime?

```
def pot(b, p):  
    y = b ** p  
    return y  
def quadrado(x):  
    a = pot(x, 2)  
    return a  
  
n = 5  
resultado = quadrado(n)  
print(resultado)
```

- a. 25
- b. 5
- c. 125
- d. 32

8. O que o seguinte código imprime?

```
def pot(b, p):  
    y = b ** p  
    return y  
def quadrado(x):  
    a = pot(x, 2)  
    return a  
  
n = 5  
resultado = quadrado(n)  
print(resultado)
```

a. 25

b. 5

c. 125

d. 32

9. O que o seguinte código imprime?

```
def adicionaNumero(lista, elem):  
    lista.append(elem)  
  
lista = [5]  
adicionaNumero(lista, 10)  
adicionaNumero(lista, 5)  
print(lista)
```

- a. Não irá compilar.
- b. [5]
- c. [5, 10]
- d. [5, 10, 5]

9. O que o seguinte código imprime?

```
def adicionaNumero(lista, elem):  
    lista.append(elem)  
  
lista = [5]  
adicionaNumero(lista, 10)  
adicionaNumero(lista, 5)  
print(lista)
```

- a. Não irá compilar.
- b. [5]
- c. [5, 10]
- d. [5, 10, 5]

10. O que o seguinte código imprime?

```
def adicionaNumero(lista, elem):  
    lista.append(elem)  
  
lista1 = [1, 2, 3, 4, 5]  
lista2 = [10, 20]  
lista1 = lista2  
adicionaNumero(lista1, 30)  
print(lista1)
```

- a. Não irá compilar.
- b. [1, 2, 3, 4, 5, 30]
- c. [10, 20, 30]
- d. [1, 2, 3, 4, 5, 10, 20, 30]

10. O que o seguinte código imprime?

```
def adicionaNumero(lista, elem):  
    lista.append(elem)  
  
lista1 = [1, 2, 3, 4, 5]  
lista2 = [10, 20]  
lista1 = lista2  
adicionaNumero(lista1, 30)  
print(lista1)
```

- a. Não irá compilar.
- b. [1, 2, 3, 4, 5, 30]
- c. [10, 20, 30]**
- d. [1, 2, 3, 4, 5, 10, 20, 30]

11. O que é impresso pelos seguintes comandos?

```
def func():  
    a = b + 10  
    return a  
  
a = 10  
b = 20  
c = func()  
print("c = ", a + b + c)
```

- a. c = 30
- b. c = 40
- c. c = 60
- d. Erro.

11. O que é impresso pelos seguintes comandos?

```
def func():  
    a = b + 10  
    return a  
  
a = 10  
b = 20  
c = func()  
print("c = ", a + b + c)
```

- a. c = 30
- b. c = 40
- c. c = 60**
- d. Erro.

12. O que o seguinte código imprime?

```
def func():  
    global a  
    b = 6  
    a = b + 30  
  
func()  
a = 10  
b = 20  
print("c = ", a + b)
```

- a. c = 30
- b. c = 56
- c. c = 86
- d. c = 36

12. O que o seguinte código imprime?

```
def func():  
    global a  
    b = 6  
    a = b + 30  
  
func()  
a = 10  
b = 20  
print("c = ", a + b)
```

- a. c = 30
- b. c = 56
- c. c = 86
- d. c = 36

13. O que será exibido pelo programa?

```
def func(p):  
    global a  
    a = b + 30  
    print("res = ", p + a)
```

```
a = 10  
func(a)  
b = 20
```

- a. res = 60
- b. res = 40
- c. res = 30
- d. Não irá compilar.

13. O que será exibido pelo programa?

```
def func(p):  
    global a  
    a = b + 30  
    print("res = ", p + a)
```

```
a = 10  
func(a)  
b = 20
```

- a. res = 60
- b. res = 40
- c. res = 30
- d. Não irá compilar.**

14. O que será exibido pelo programa?

```
mat = []
n = 2
for i in range(n):
    lista = []
    for j in range(n):
        lista.append(1*i)
    mat.append(lista)
print(mat)
```

- a. `[[1, 1], [2, 2]]`
- b. `[[0, 0], [1, 1]]`
- c. `[]`
- d. `[[0, 1], [0, 1]]`

14. O que será exibido pelo programa?

```
mat = []
n = 2
for i in range(n):
    lista = []
    for j in range(n):
        lista.append(1*i)
    mat.append(lista)
print(mat)
```

- a. `[[1, 1], [2, 2]]`
- b. `[[0, 0], [1, 1]]`**
- c. `[]`
- d. `[[0, 1], [0, 1]]`

15. Quantas comparações são feitas na Busca Sequencial e na Busca Binária até o valor da posição onde se encontra a chave a ser retornada?

```
lista = [2, 5, 6, 7, 9, 10]  
chave = 9
```

- a. Sequencial: 6 comparações & Binária: 3 comparações
- b. Sequencial: 3 comparações & Binária: 6 comparações
- c. Sequencial: 5 comparações & Binária: 2 comparações
- d. Sequencial: 2 comparações & Binária: 5 comparações

15. Quantas comparações são feitas na Busca Sequencial e na Busca Binária até o valor da posição onde se encontra a chave a ser retornada?

```
lista = [2, 5, 6, 7, 9, 10]  
chave = 9
```

- a. Sequencial: 6 comparações & Binária: 3 comparações
- b. Sequencial: 3 comparações & Binária: 6 comparações
- c. Sequencial: 5 comparações & Binária: 2 comparações**
- d. Sequencial: 2 comparações & Binária: 5 comparações

16. Para a lista $[3, 5, 1, 2, 0, 4]$, qual foi o algoritmo de ordenação aplicado?

```
[3, 5, 1, 2, 0, 4]
[3, 1, 5, 2, 0, 4]
[3, 1, 2, 5, 0, 4]
[3, 1, 2, 0, 5, 4]
[3, 1, 2, 0, 4, 5]
...
[0, 1, 2, 3, 4, 5]
```

- a. `selectionSort(lista)`
- b. `bubbleSort(lista)`
- c. `insertionSort(lista)`
- d. Nenhuma das opções acima.

16. Para a lista $[3, 5, 1, 2, 0, 4]$, qual foi o algoritmo de ordenação aplicado?

```
[3, 5, 1, 2, 0, 4]
[3, 1, 5, 2, 0, 4]
[3, 1, 2, 5, 0, 4]
[3, 1, 2, 0, 5, 4]
[3, 1, 2, 0, 4, 5]
...
[0, 1, 2, 3, 4, 5]
```

- a. `selectionSort(lista)`
- b. `bubbleSort(lista)`**
- c. `insertionSort(lista)`
- d. Nenhuma das opções acima.

17. Para a lista `[3, 5, 1, 2, 0, 4]`, qual foi o algoritmo de ordenação aplicado?

```
[0, 5, 1, 2, 3, 4]
```

```
[0, 1, 5, 2, 3, 4]
```

```
[0, 1, 2, 5, 3, 4]
```

```
[0, 1, 2, 3, 5, 4]
```

```
[0, 1, 2, 3, 4, 5]
```

```
[0, 1, 2, 3, 4, 5]
```

- a. `selectionSort(lista)`
- b. `bubbleSort(lista)`
- c. `insertionSort(lista)`
- d. Nenhuma das opções acima.

17. Para a lista `[3, 5, 1, 2, 0, 4]`, qual foi o algoritmo de ordenação aplicado?

```
[0, 5, 1, 2, 3, 4]
```

```
[0, 1, 5, 2, 3, 4]
```

```
[0, 1, 2, 5, 3, 4]
```

```
[0, 1, 2, 3, 5, 4]
```

```
[0, 1, 2, 3, 4, 5]
```

```
[0, 1, 2, 3, 4, 5]
```

- a. `selectionSort(lista)`
- b. `bubbleSort(lista)`
- c. `insertionSort(lista)`
- d. Nenhuma das opções acima.

18. Para a lista $[3, 5, 1, 2, 0, 4]$, qual foi o algoritmo de ordenação aplicado?

$[3, 5, 1, 2, 0, 4]$

$[1, 3, 5, 2, 0, 4]$

$[1, 2, 3, 5, 0, 4]$

$[0, 1, 2, 3, 5, 4]$

$[0, 1, 2, 3, 4, 5]$

- a. `selectionSort(lista)`
- b. `bubbleSort(lista)`
- c. `insertionSort(lista)`
- d. Nenhuma das opções acima.

18. Para a lista $[3, 5, 1, 2, 0, 4]$, qual foi o algoritmo de ordenação aplicado?

$[3, 5, 1, 2, 0, 4]$

$[1, 3, 5, 2, 0, 4]$

$[1, 2, 3, 5, 0, 4]$

$[0, 1, 2, 3, 5, 4]$

$[0, 1, 2, 3, 4, 5]$

- a. `selectionSort(lista)`
- b. `bubbleSort(lista)`
- c. `insertionSort(lista)`**
- d. Nenhuma das opções acima.

19. O que é impresso pelo trecho de código a seguir?

```
def rec(n):  
    if n == 10:  
        return 1  
    return 1 + rec(n+1)  
  
n = 6  
print(rec(n))
```

- a. 7
- b. 5
- c. 6
- d. 1

19. O que é impresso pelo trecho de código a seguir?

```
def rec(n):  
    if n == 10:  
        return 1  
    return 1 + rec(n+1)  
  
n = 6  
print(rec(n))
```

- a. 7
- b. 5**
- c. 6
- d. 1

Questão 1

(Prova 2018/1)

Questão 1 (a)

Programa	O que será exibido na tela?
<pre># (0.2 ponto) def inverte_sinal(a): print("-a =", -a) inverte_sinal(-3)</pre>	

Questão 1 (a)

Programa	O que será exibido na tela?
<pre># (0.2 ponto) def inverte_sinal(a): print("-a =", -a) inverte_sinal(-3)</pre>	<p>-a = 3</p>

Questão 1 (b)

Programa	O que será exibido na tela?
<pre># (0.3 ponto) def inverte_sinal(a): a = -a return a a = inverte_sinal(5) print("-a =", -a)</pre>	

Questão 1 (b)

Programa	O que será exibido na tela?
<pre># (0.3 ponto) def inverte_sinal(a): a = -a return a a = inverte_sinal(5) print("-a =", -a)</pre>	<p style="text-align: center;">-a = 5</p>

Questão 1 (c)

Programa	O que será exibido na tela?
<pre># (0.3 ponto) def inverte_sinal(a): a = -a inverte_sinal(8) print("-a =", -a)</pre>	

Questão 1 (c)

Programa	O que será exibido na tela?
<pre># (0.3 ponto) def inverte_sinal(a): a = -a inverte_sinal(8) print("-a =", -a)</pre>	<p>Erro na linha 5 (<code>print("-a =", -a)</code>), a não foi definido.</p>

Programa

```
# (0.3 ponto)
def inverte_sinal(a):
    a = -a
    print("a invertido = ", a)

a = -1
inverte_sinal(a)
inverte_sinal(a)
inverte_sinal(a)
if a == 1:
    print("Ficou invertido.")
elif a == -1:
    print("Não ficou invertido.")
else:
    print("-a =", -a)
```

O que será exibido na tela?

Programa	O que será exibido na tela?
<pre># (0.3 ponto) def inverte_sinal(a): a = -a print("a invertido = ", a) a = -1 inverte_sinal(a) inverte_sinal(a) inverte_sinal(a) if a == 1: print("Ficou invertido.") elif a == -1: print("Não ficou invertido.") else: print("-a =", -a)</pre>	<pre>a invertido = 1 a invertido = 1 a invertido = 1 Não ficou invertido.</pre>

Questão 1 (e)

Programa	O que será exibido na tela?
<pre># (0.4 ponto) def inverte_lista(l): for i in range(1,l): l[i] = -l[i] print(l) return l l = [5, 4, 3] inverte_lista(l.copy()) print(l)</pre>	

Questão 1 (e)

Programa	O que será exibido na tela?
<pre># (0.4 ponto) def inverte_lista(l): for i in range(1,l): l[i] = -l[i] print(l) return l l = [5, 4, 3] inverte_lista(l.copy()) print(l)</pre>	<p>Erro na linha 2 (<code>for i in range(1,l):</code>), não é possível iterar sobre a lista.</p> <p>O correto seria:</p> <pre>for i in range(1,len(l)):</pre>

Questão 1 (f)

Programa	O que será exibido na tela?
<pre># (0.5 ponto) def inverte_misterio(m): misterio_m = m.copy() for i in range(len(m)): misterio_m[i][0] = -m[i][0] print(misterio_m) m = [[-1, 3, 5], [6, 3, 2], [7, -2, 0]] inverte_misterio(m) print(m)</pre>	

Questão 1 (f)

Programa	O que será exibido na tela?
<pre># (0.5 ponto) def inverte_misterio(m): misterio_m = m.copy() for i in range(len(m)): misterio_m[i][0] = -m[i][0] print(misterio_m) m = [[-1, 3, 5], [6, 3, 2], [7, -2, 0]] inverte_misterio(m) print(m)</pre>	<pre>[[1, 3, 5], [-6, 3, 2], [-7, -2, 0]] [[1, 3, 5], [-6, 3, 2], [-7, -2, 0]]</pre>

Questão 3

(Prova 2018/1)

Questão 3 (a)

Como exemplo de chamada da função `busca_sequencial`, considere o seguinte programa:

`programa_item_c.py`

```
lista = [11, 19, 12, 13, 19, 43, 32, 41, 11, 12, 24]
chave = 12
pos = busca_sequencial(lista, chave)
if (pos == -1):
    print("Não existe")
else:
    print("Chave localizada na posição = ", pos)
```


Questão 3 (a)

(0.8 ponto) Implemente aqui a função `busca_sequencial`:

A large empty rectangular box with a thin black border, intended for the implementation of the `busca_sequencial` function.

Questão 3 (a)

(0.8 ponto) Implemente aqui a função `busca_sequencial`:

```
def busca_sequencial(lista, chave):  
    for i in range(len(lista)):  
        if lista[i] == chave:  
            return i  
    return -1
```

Podemos ter várias soluções.
O programa apresentado é uma
possível resposta.

Questão 3 (b)

(0.6 ponto) Você considera que é possível implementar a Busca Binária na situação descrita no item a? Ou seja, a 3a linha do `programa_item_c.py` seria substituída por: `pos = busca_binaria(lista, chave)`.

Responda SIM ou NÃO e **justifique** a sua escolha.

Questão 3 (b)

(0.6 ponto) Você considera que é possível implementar a Busca Binária na situação descrita no item a? Ou seja, a 3a linha do `programa_item_c.py` seria substituída por: `pos = busca_binaria(lista, chave)`.

Responda SIM ou NÃO e **justifique** a sua escolha.

NÃO, porque para a busca binária funcionar a lista precisa estar ordenada, e a lista não está ordenada.

(0.8 ponto) Vamos ajudar Mário a compreender o algoritmo de Busca Binária? Explique aqui (use português, não código de programação) o algoritmo da busca binária. Indique (e explique!) se João estava certo ao afirmar que a busca binária é mais eficiente que a busca sequencial.

(0.8 ponto) Vamos ajudar Mário a compreender o algoritmo de Busca Binária? Explique aqui (use português, não código de programação) o algoritmo da busca binária. Indique (e explique!) se João estava certo ao afirmar que a busca binária é mais eficiente que a busca sequencial.

A busca binária parte do pressuposto de que a lista está ordenada. A ideia é a seguinte: primeiro verifique se a chave de busca é igual ao valor da posição do meio da lista. Caso seja igual, devolva esta posição. Caso o valor desta posição seja maior, então repita o processo mas considere que a lista tem metade do tamanho, indo até posição anterior a do meio. Caso o valor desta posição seja menor, então repita o processo mas considere que a lista tem metade do tamanho e inicia na posição seguinte a do meio.

João estava certo. A busca binária é mais eficiente que a busca sequencial porque faz menos comparações, uma vez que a cada iteração o espaço de busca está sendo reduzido pela metade.

Podemos ter várias respostas para essa questão.

Questão 3 (d)

(1.0 ponto) Considere a seguinte lista $[3, 5, 7, 9, 11, 12, 13, 14]$. A chave a ser procurada é o **número 3**. Complete cada uma das iterações da busca binária, seguindo o modelo abaixo. ... Ao final, indique se a chave procurada foi ou não encontrada. Em caso positivo, qual é a sua posição na lista?

	3	5	7	9	11	12	13	14
Iteração 1	I			M				F
Iteração 2	I	M	F					
Iteração 3	IMF							

A chave foi encontrada na posição 0 (zero).

Questão 4

(Prova 2018/1)

Questão 4

(3.0 pontos) Continuando seus estudos sobre funções, João implementou a seguinte função recursiva:

```
1: def func(n):  
2:     print("n = ", n)  
3:     if n == 0 or n == 1:  
4:         return 1  
5:     return n * func(n-1)
```

Questão 4 (a)

(0.4 ponto) O que o código `print("r = ", func(4))` irá imprimir na tela quando executado?

Questão 4 (a)

(0.4 ponto) O que o código `print("r = ", func(4))` irá imprimir na tela quando executado?

```
n = 4
n = 3
n = 2
n = 1
r = 24
```

Questão 4 (b)

(0.2 ponto) Do ponto de vista matemático, pode-se afirmar que a função implementada por João resolve qual problema?

Questão 4 (b)

(0.2 ponto) Do ponto de vista matemático, pode-se afirmar que a função implementada por João resolve qual problema?

Fatorial.

Questão 4 (c)

(0.4 ponto) Caso as linhas 3 e 4 fossem removidas da função, o que ocorreria quando a mesma fosse executada?

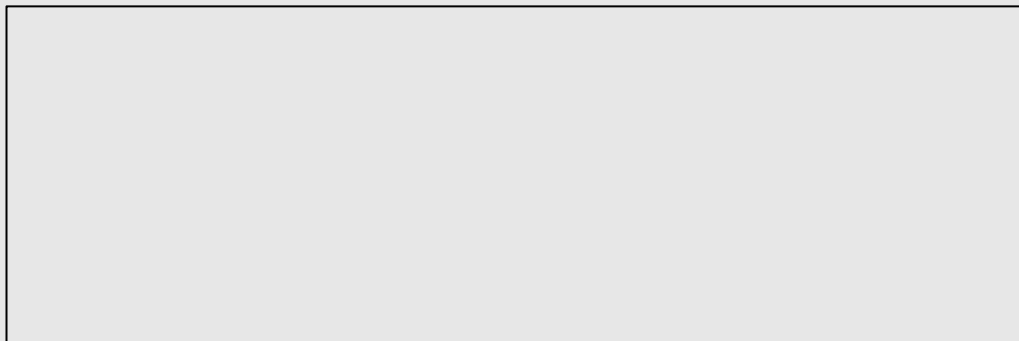
Questão 4 (c)

(0.4 ponto) Caso as linhas 3 e 4 fossem removidas da função, o que ocorreria quando a mesma fosse executada?

O programa entra em loop infinito.

Questão 4 (d)

(1.0 ponto) Implemente a função *func* com uma solução iterativa (sem usar recursão).

A large, empty rectangular box with a thin black border, intended for the student to write their iterative implementation of the function *func*.

Questão 4 (d)

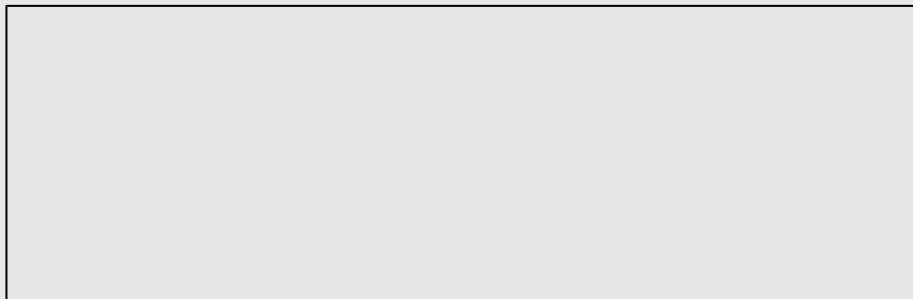
(1.0 ponto) Implemente a função *func* com uma solução iterativa (sem usar recursão).

```
def func(n):  
    total, k = 1, 1  
    while k <= n:  
        total, k = total * k, k + 1  
    return total
```

Podemos ter várias soluções.
O programa apresentado é uma possível resposta.

Questão 4 (e)

(1.0 ponto) Implemente uma função recursiva chamada `soma_n_naturais(n)` que recebe como parâmetro um número natural n ($n \geq 0$) e retorna a soma de todos os naturais de 0 até n (inclusive). Por exemplo, a função irá retornar: 0 se $n == 0$; 1 se $n == 1$; 3 se $n == 2$; 6 se $n == 3$ e assim por diante.



Questão 4 (e)

(1.0 ponto) Implemente uma função recursiva chamada `soma_n_naturais(n)` que recebe como parâmetro um número natural n ($n \geq 0$) e retorna a soma de todos os naturais de 0 até n (inclusive). Por exemplo, a função irá retornar: 0 se $n == 0$; 1 se $n == 1$; 3 se $n == 2$; 6 se $n == 3$ e assim por diante.

```
def soma_n_naturais(n):  
    print("n = ", n)  
    if n == 0:  
        return 0  
    return n + soma_n_naturais(n-1)
```

Podemos ter várias soluções.
O programa apresentado é uma possível resposta.

Exercícios

- <https://wiki.python.org.br/ListaDeExercicios>

**Boa
Prova!**

AMO
PROGRAMAR



! ! !
D - - -
X - - -
T - - -
- - -
o - - -
n - - -
! ! !