

**Lista 14****Busca binária; Recursão;****Questão 1**

Assinale as alternativas **corretas** sobre recursão:

- (a) Base da recursão é a condição que faz o problema ser definitivamente resolvido. Caso essa condição, essa base da recursão, não seja satisfeita, o problema continua sendo reduzido em instâncias menores até que a condição passe a ser satisfeita.
- (b) Chamada recursiva é a linha onde é chamada pela primeira vez a função recursiva.
- (c) Chamada recursiva é a linha onde a função faz uma chamada a ela mesma.
- (d) Função recursiva é a função que chama ela mesma.
- (e) Base da recursão é o conjunto de reduções do problema em instâncias menores.

**Questão 2**

Observe o código abaixo:

```
1 def fatorial(n):  
2     if n <= 1:  
3         return n  
4     else:  
5         return n * fatorial(n-1)
```

Assinale as afirmações **corretas**:

- (a) A linha 2 tem a condição que é a base da recursão
- (b) O algoritmo está correto
- (c) Para o algoritmo funcionar corretamente, é necessário trocar a linha 3 por `return 1`
- (d) A linha 5 tem a condição que é a base da recursão
- (e) A linha 5 tem a chamada recursiva
- (f) A linha 2 tem a chamada recursiva

---

### Questão 3

Observe o código abaixo:

```
1 def fibo(n):  
2     <base da recursão>  
3     return n  
4 else:  
5     return fibo(n-1) + fibo(n-2)
```

A linha 2 deve ter a condição que representa a base da recursão desse algoritmo.

Assinale as linhas que poderiam substituir a linha 2, sendo que o algoritmo deve retornar o cálculo do e-nésimo valor da série Fibonacci:

- (a) `if (n < 2):`
- (b) `if (n == 1):`
- (c) `if (n >= 0):`
- (d) `if (n == 2):`
- (e) `if (n <= 1):`

### Questão 4

Observe o código abaixo:

```
1 def x(n):  
2     if n == 0:  
3         <espaço A>  
4     else:  
5         <espaço B>  
6         x(n-1)  
7         <espaço C>  
8     <espaço D>  
9 <espaço E>
```

Se a função acima for chamada como `x(10)`, onde é necessário colocar o comando `print(n)` para que a função imprima os números de 0 até 10?

- (a) No <espaço A> e no <espaço B>
- (b) No <espaço A>, no <espaço B> e no <espaço C>
- (c) No <espaço A> e no <espaço C>
- (d) No <espaço B> e no <espaço C>
- (e) No <espaço C> e no <espaço E>
- (f) No <espaço A> e no <espaço E>
- (g) Nenhuma das opções apresentadas está correta

---

## Questão 5

Observe o código abaixo:

```
1 def x(n, m):
2     if n == m or m == 0:
3         return 1
4     else:
5         return x(n-1,m) + x(n-1,m+1)
6
7 print(x(5,3))
```

Qual será o resultado e quantas chamadas recursivas serão realizadas?

- (a) 66 e 8
- (b) 34 e 12
- (c) 43 e 6
- (d) Looping infinito
- (e) Nenhuma das alternativas mostra a resposta correta

## Questão 6

Observe o código abaixo:

```
1 def x(n):
2     if n >= 0 and n <= 2:
3         return n
4     else:
5         return x(n-1) + x(n-2) + x(n-3)
6
7 print(x(6))
```

Qual será o resultado e quantas chamadas recursivas serão realizadas?

- (a) Resultado: 37. Chamadas recursivas: 45
- (b) Resultado: 11. Chamadas recursivas: 12
- (c) Resultado: 3. Chamadas recursivas: 3
- (d) Resultado: 6. Chamadas recursivas: nenhuma.
- (e) Resultado: 6. Chamadas recursivas: 6
- (f) Resultado: 20. Chamadas recursivas: 24

## Questão 7

Implemente a função `soma_lista(lista)`, que recebe como parâmetro uma lista de números inteiros e devolve um número inteiro correspondente à soma dos elementos desta lista.

Sua solução deve ser implementada utilizando **recursão**.

---

### Questão 8

Implemente a função `fibonacci(n)`, que recebe como parâmetro um número inteiro e devolve um número inteiro correspondente ao  $n$ -ésimo elemento da sequência de Fibonacci. Sua solução deve ser implementada utilizando recursão.

```
1 fibonacci(4)
2 # deve devolver => 3
3 fibonacci(2)
4 # deve devolver => 1
```

### Questão 9

Implemente a função `busca_binaria(x)`, que recebe como parâmetro uma lista de números inteiros, um valor  $x$ , e retorna uma posição em que  $x$  ocorre na lista, ou `False` caso contrário.

Sua solução deve ser implementada utilizando recursão.

```
1 a = [-10, -2, 0, 5, 66, 77, 99, 102, 239, 567, 875, 934]
2 busca_binaria(a, 99)
3 # deve devolver => 6
```

### Questão 10

Escreva uma **função recursiva** que receba uma matriz `A` de números inteiros e retorna o **maior valor** na matriz.

Exemplo:

```
1 M = [[4, 24, 19, 49, 19],
2       [14, 4, 27, 54, 68],
3       [82, 53, 58, 36, 73],
4       [38, 38, 58, 12, 38]]
5
6 >>> maior_valor(M)
7 >>> 82
```