

Fundamentos de Programação

Funções - Recursão

Thiago Alves

Introdução

- Faça uma função `fib(n)` que retorna o n -ésimo elemento da sequência de fibonacci para um parâmetro n .
- Representação recursiva da sequência de fibonacci?

Recursão em Python

```
def fib(n):  
    if n==1 or n==2:  
        return 1  
    return fib(n-1)+fib(n-2)
```

```
posicao = input()  
print fib(posicao)
```

Exercícios

- Faça uma função recursiva `fat(n)` para retornar o fatorial de um inteiro não negativo dado como parâmetro.

Exercícios

- Faça uma função recursiva `member(x, lista)` que tem como parâmetros um valor qualquer e uma lista, respectivamente. A função deve retornar `True`, se o valor é um dos elementos da lista e `False`, caso contrário.

Exercícios

```
def member(x, lista):  
    if lista == []:  
        return False  
  
    if x == lista[0]:  
        return True  
  
    del lista[0]  
    return member(x, lista)
```

Exercícios

- Faça uma função recursiva `produtoLista(lista)` que tem como parâmetro uma lista de números. A função deve retornar o produto dos elementos da lista.

Exercícios

- Defina uma função recursiva `somaTodos(lista, x)` que tem uma lista de números e um número como parâmetros, respectivamente. A função deve somar todos os elementos da lista por `x` e retornar a nova lista.

Exercícios

- Faça uma função recursiva `hanoi (n, Orig, Aux, Dest)` que tem como parâmetros o número de discos, o nome da torre de origem, o da torre auxiliar e o da torre de destino do problema das torres de Hanói. A função deve retornar a quantidade de movimentos para mover todos os discos da torre de origem para a de destino de acordo com as restrições do problema.