DISCIPLINA: Estrutura de Dados | PROFESSOR: Douglas GC

ATIVIDADE DE RECURSÃO EM PYTHON

SUGESTÃO: use o Google Colab clicando no link a seguir https://colab.research.google.com/drive/1sA3nOf0ruoDQ2Oanl0aNfYajPKerFQXe?usp=sharing

⭐▶ ***INSTRUÇÃO:*** Neste exercício iremos trabalhar um conceito muito utilizado em estrutura de dados. Trata-se da "Recursão". A recursão ocorre quando uma função chama a ela mesma. Mas, oberve que se uma função chamar a si mesma indefinidamente se tornará um loop infinito, ou seja, um erro de programação. Portanto, existe a necessidade de programar um ponto de parada ou condição de parada.

✍▶ A atividade de hoje consiste em programar duas funções diferentes para cada exercício. A primeira função é a mais simples, isto é, sem recursão. A segunda função deve apresentar o mesmo resultado da primeira, porém, usando recursão.

Então, mãos à obra e bons estudos!! ✌ 😊

NOTA: *este exercício ZERO será realizado pelo professor com o objetivo de transmitir o conceito de recursividade.*

EXERCÍCIO 0: dada a entrada de uma palavra "P" e um número "X", a função deve repetir P X vezes.

```
# FUNÇÃO SEM RECURSIVIDADE

def repetidorS(p, x):
  for n in range(x):
    print(p)

palavra = input("Digite a palavra que será repetida: ")
vezes = int(input("Repetir quantas vezes? "))

repetidorS(palavra, vezes)
```

```
Digite a palavra que será repetida: Uva
Repetir quantas vezes? 3
Uva
Uva
Uva
```

```
# FUNÇÃO COM RECURSIVIDADE

def repetidorR(p, x):
  if x == 0:
    return

  print(p)
  return repetidorR(p, x - 1)

palavra = input("Digite a palavra que será repetida: ")
vezes = int(input("Repetir quantas vezes? "))

repetidorR(palavra, vezes)
```

```
Digite a palavra que será repetida: Uva
Repetir quantas vezes? 3
Uva
Uva
Uva
```

EXERCÍCIO 01: função que solicita ao usuário a entrada de um número qualquer. Essa função deve somar todos os números em decremento de 1. Por fim, a função retorna o resultado do cálculo. Por exemplo, a entrada do valor 5 tem como resultado 5+4+3+2+1=15.

```python
# FUNÇÃO SEM RECURSIVIDADE

def somaN(n):
  soma = 0
  for i in range(n + 1):
    soma += i
  return soma

print(somaN(2))
```

```
3
```

```python
# FUNÇÃO COM RECURSIVIDADE

def somaR(n):
  return n + somaN(n - 1) if n != 0 else 0

print(somaN(2))
```

```
3
```

EXERCÍCIO 02: calcular fatorial de um número fornecido pelo usuário. Por fim, a função retorna o resultado do cálculo.

```python
def fatorialS(n):
  fatorial = 1
  for x in range(n):
    fatorial = fatorial * (x + 1)
  return fatorial

print(fatorialS(3))
```

```
6
```

```python
def fatorialR(n):
  if n == 0:
    return 1
  return n * fatorialR(n-1)

print(fatorialR(5))
```

```
120
```

EXERCÍCIO 03: a função solicita uma lista de valores do usuário. Em seguida, a função deve somar todos os valores dessa lista. Por fim, a função retorna o resultado da soma.

```python
listaValores = [1, 2, 3, 4, 5, 10]

def somarLista(listaSoma, i=0):
  if len(listaSoma) - i == 0:
    return 0
  i = i + 1
  return listaSoma[len(listaValores) - i] + somarLista(listaSoma, i)

print(somarLista(listaValores))

```

25

EXERCÍCIO 04: dada uma lista de valores fornecidas pelo usuário, a função deve retornar a mesma lista, porém, com a ordem de valores invertida (de trás para frente).

```
listaValores = [1,2,30,400,5000]
listaInvert = []

def inverterLista(listaValores, listaInvert, i=1):

  if len(listaValores) - i == -1:
    return

  listaInvert.append(listaValores[len(listaValores) - i])
  i = i + 1
  inverterLista(listaValores, listaInvert, i)
```

```
inverterLista(listaValores, listaInvert)
print(listaInvert)
```

```
[5000, 400, 30, 2, 1]
```