

Exercícios de recursão aula de 17/04/2018:

Implementar em Python as seguintes funções recursivas e escrever o programa que testa a função.

- 1) Escrever um número inteiro na vertical. Parâmetro: número inteiro

```
def escrever_num_vertical(n):
    if n==0:
        return
    escrever_num_vertical(n//10)
    print(n%10)
    return

escrever_num_vertical(54321)
```

- 2) Escrever uma string na vertical. Parâmetro: string

```
def escrever_string_vertical(s):
    if s=="":
        return
    print(s[0])
    return escrever_string_vertical(s[1:])

escrever_string_vertical('abcde')
```

- 3) Escrever um número inteiro invertido. Parâmetro: número inteiro

```
#Solução 1
def escrever_num_vertical_inv(n):
    if n==0:
        return
    print(n%10)
    escrever_num_vertical_inv(n//10)
    return

escrever_num_vertical_inv(54321)
```

```
#Solução 2 (Desafio)
def num_inv(n,m):
    if n==0:
        return 0
    r=n%10
    d=n//10
```

```
x= num_inv(d,m-1)
return r*10**(m-1)+x
```

```
print(num_inv(54321,5))
```

OBS: Essa solução funciona com apenas 1 parâmetro se for usada a função do exercício 6 para contar o número de dígitos do número inteiro para depois chamar a função num_inv() com 2 parâmetros.

- 4) Escrever uma string invertida. Parâmetro: string

```
def string_inv(s):
    if s=="":
        return ""
    return string_inv(s[1:])+s[0]
```

```
print(string_inv('abcde'))
```

- 5) Verificar se uma string é um palíndromo, retornando False ou True. Parâmetro: string

```
def pal(s):
    if len(s)==1 or s=="":
        return True
    m=len(s)
    if s[0]==s[m-1]:
        return pal(s[1:m-1])
    else:
        return False
```

```
print(pal('socorrammesubinoonibusemmarrocoss'))
print(pal('natan'))
print(pal('alo'))
print(pal('abcd'))
```

- 6) Contar e retornar o número de dígitos em um inteiro. Parâmetro: número inteiro

```
def contar_dig_int(n):
    if n==0:
        return 0
    return(1+contar_dig_int(n//10))
```

```
print(contar_dig_int(7654321))
```

- 7) Contar e retornar o número de letras em uma string. Parâmetro: string

```
def contar_car_str(s):
    if s=="":
        return 0
    if 'A'<=s[0]<='Z' or 'a'<=s[0]<='z':
        return 1 + contar_car_str(s[1:])
    else:
        return contar_car_str(s[1:])

print(contar_car_str('abcdefgh-+'))
print(contar_car_str(""))
print(contar_car_str('!@#$'))
```

- 8) Contar e retornar o número de vezes em que aparece uma determinada letra em uma string. Parâmetros: string e a letra a ser contada

```
def contar_car_str2(s,c):
    if s=="":
        return 0
    if c==s[0]:
        return 1 + contar_car_str2(s[1:],c)
    else:
        return contar_car_str2(s[1:],c)

print(contar_car_str2('abcdefgh-+', '+'))
print(contar_car_str2('', '1'))
print(contar_car_str2('123123123', '1'))
```

- 9) Contar e retornar o número de vezes em que aparece um determinado dígito em um número inteiro. Parâmetros: número inteiro e o dígito (inteiro) a ser contado

```
def contar_dig_int(n,d):
    if n==0:
        return 0
    if d==n%10:
        return 1+contar_dig_int(n//10,d)
    else:
        return contar_dig_int(n//10,d)

print(contar_dig_int(7555351,5))
print(contar_dig_int(3,3))
print(contar_dig_int(321,5))
```