

A resolução desta atividade corresponde a uma questão de prova (01/01/2024) e valerá 1 ponto.

**ROBERTHA DE SOUSA REZENDE**

1. Descreva os custos e benefícios de definir e utilizar uma função recursiva.

R= A função recursiva torna o código mais claro e intuitivos, embora não tem nenhum benefício quanto ao desempenho. Além disso, essa função torna os problemas em subproblemas.

2. Explique o que acontece quando a seguinte função recursiva é chamada com o valor 4 como argumento:

R= O número 4 passa pela verificação (chamada condicional do código) e ele é MAIOR que 0. Então ele vai imprimir o número 4 e chamar o próximo número em ordem decrescente, ou seja, o 3 será o próximo número. Será assim até chegar no número 1.

```
def example(n):  
    if n > 0:  
        print(n)  
        example(n - 1)
```

3. Explique o que acontece quando a seguinte função recursiva é chamada com o valor 4 como argumento:

R= Nesse caso, essa função segue as regras da recursividade, tendo o caso base, caso recursivo e chama a si mesma também. Na condição, se o número quatro for maior que zero, a função imprime o número e chama a si mesma, pois não houve a diminuição do n.

```
def example(n):  
    if n > 0:  
        print(n)  
        example(n)  
    else:  
        example(n - 1)
```

4. Uma função é recursiva quando ela atende as seguintes leis:

1. Um algoritmo recursivo deve ter um *caso básico*

R= o caso base serve como ponto de parada da função, serve como um critério de quando a função vai ter que parar, e não chama a si mesma.

Exemplo:

```
def fat(numero):  
    if numero == 0:  
        return 1  
    else:  
        return numero*fat(numero-1)  
numero=int(input('Informe um numero inteiro: '))  
print(fat(numero))
```

2. Um algoritmo recursivo deve mudar o seu estado e se aproximar do caso básico.  
R= é quando uma função modifica suas configurações para que ela retorne ao caso base.

Exemplo:

```
def soma(n):  
    if n == 1:  
        return 1  
    else:  
        return n + soma(n - 1)  
  
n=int(input('Informe um número: '))  
print(soma(n))
```

3. Um algoritmo recursivo deve chamar a si mesmo, recursivamente.  
R= é quando uma função chama a si mesma para resolver os subproblemas, até retornar para o caso base.

Exemplo:

```
def recursiva(fibon):  
    if fibon == 1:  
        return 0  
    elif fibon == 2:  
        return 1  
    else:  
        return recursiva(fibon- 1) + recursiva(fibon-2)  
fibon= 10  
print(recursiva(fibon))
```