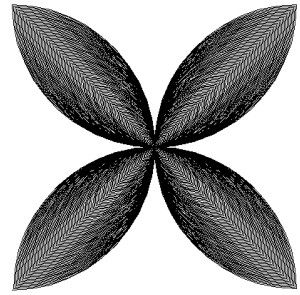
DEPARTAMENTO DE INFORMÁTICA PUC-RIO

Recursão

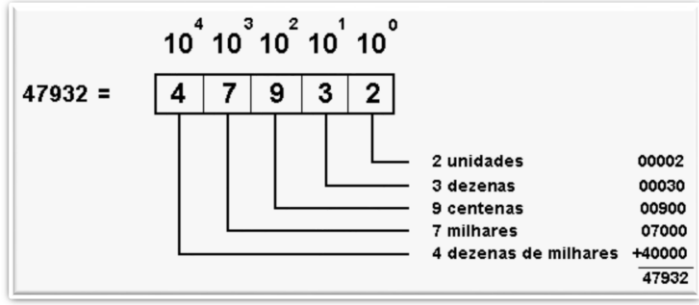


DEPARTAMENTO DE INFORMÁTICA PUC-RIO

Interpretação Recursiva de um Número Inteiro

Sistema decimal

- ✓ Número é uma sequência de algarismos
- ✓ O valor de cada algarismo depende da posição que ele ocupa
  - sistema posicional
- ✓ Da direita para a esquerda, cada posição determina a multiplicação do algarismo por uma potência de 10

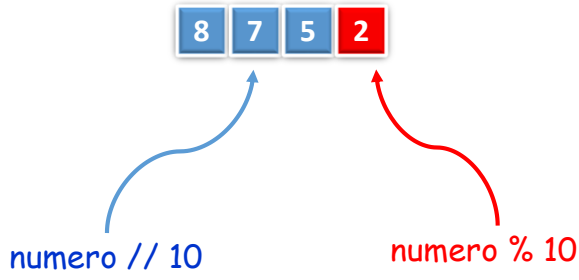


3

Interpretação recursiva

- ✓ Um número inteiro no sistema decimal é
  - um algarismo

    OU
  - um algarismo precedido por um número
- ✓ Exemplo



4

Exemplo

- ✓ Crie uma função recursiva chamada ***somaAlg*** que retorne a soma dos algarismos de um número inteiro positivo recebido como parâmetro
- ✓ Exemplo: 323 ➔ 3 + 2 + 3 ➔ 8
- ✓ Caso base?
  - número < 10 ➔ a soma é o próprio número
- ✓ Passo recursivo
  - unidade + somaAlg (numero // 10)

```
somaAlg(n) { n < 10:  n
           { demais casos: unidade + soma(n // 10)
```

5

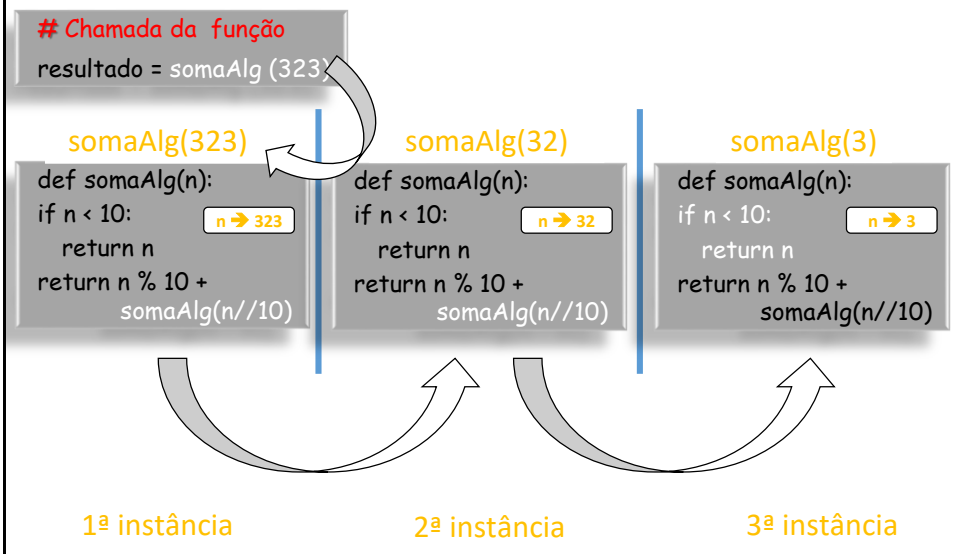
Função *somaAlg*

```
def somaAlg (n):
    if n < 10:
      return n
    return n % 10 + somaAlg(n // 10)

# Chamada da  função
resultado = somaAlg (323)
print ('Soma dos algarismos = %d' %resultado)
```

6

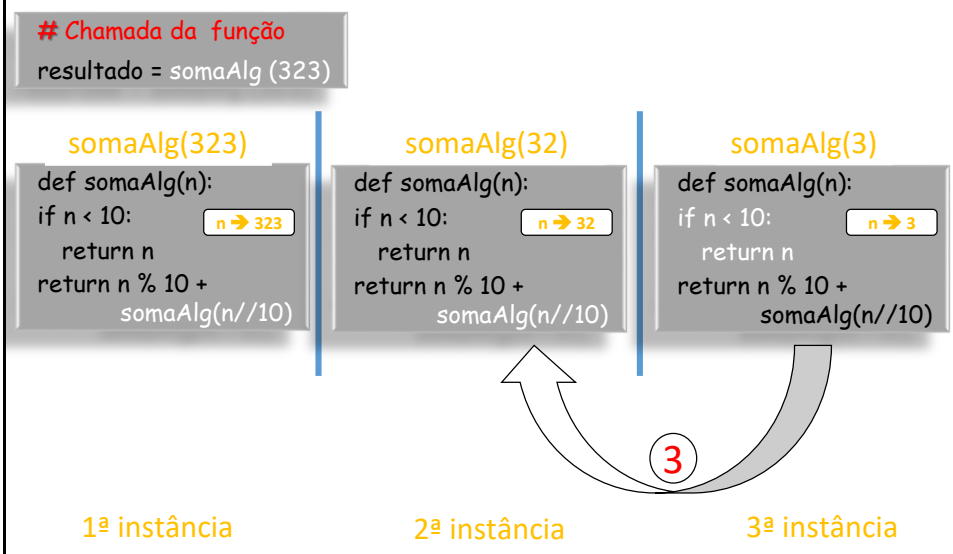
Simulação da execução



1ª instância 2ª instância 3ª instância

7

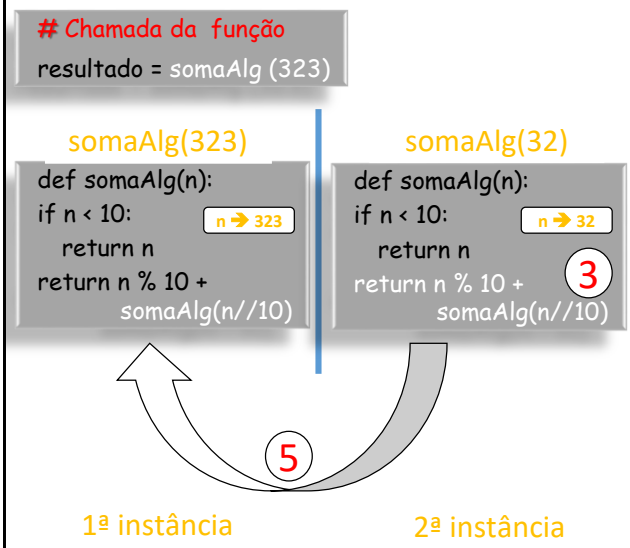
Simulação da execução



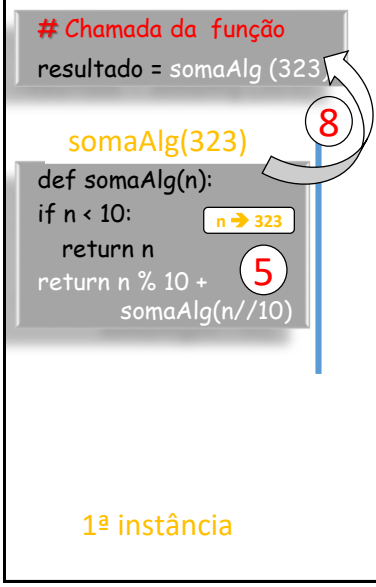
1ª instância 2ª instância 3ª instância

8

Simulação da execução



Simulação da execução



Simulação da execução

Chamada da função 8

resultado = somaAlg (323)

✓ Na variável resultado, será armazenado o valor 8

Exemplo

- ✓ Crie uma função recursiva chamada **exibeVert** que exiba verticalmente os algarismos de um número positivo recebido como parâmetro
- ✓ Exemplo: 875 ➔ 8
  7
  5
- ✓ Caso base?
  - número < 10 ➔ exibe o próprio número
- ✓ Passo recursivo
  - exibeVert (numero // 10) e exibe unidade

numero % 10

Exemplo

- ✓ Algoritmo
  - se o número só tiver um dígito, imprime o número e retorna
  - senão, faz a chamada recursiva passando o número sem a unidade e, após o retorno da função, imprime a unidade
    - a chamada recursiva imprimirá o número sem a unidade

```
exibeVert(n)  { n < 10: exibe n
              { caso contrário: exibeVert(n//10)
                                + exibe unidade
```

13

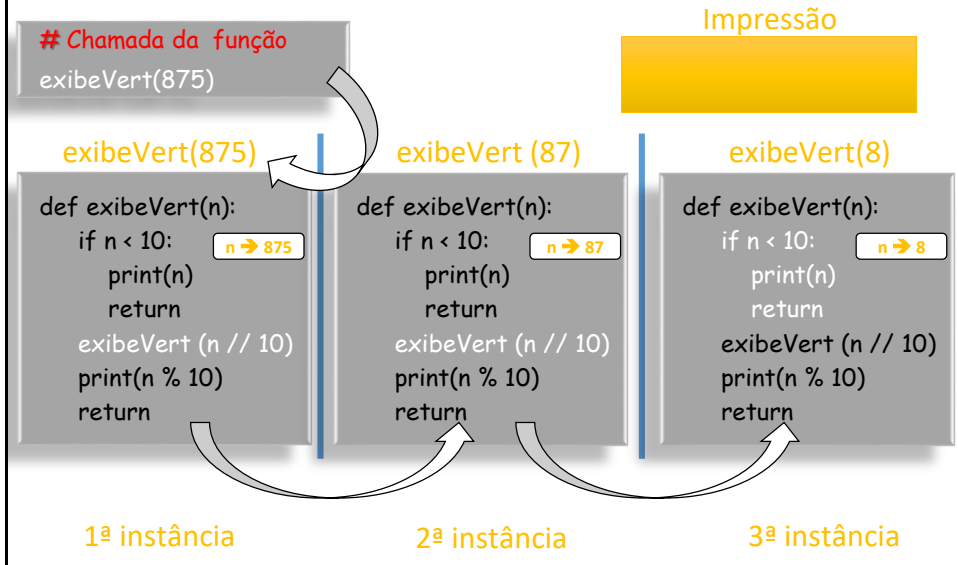
Função *exibeVert*

```
def exibeVert (n):
    if n < 10:
      print(n)
      return
    exibeVert (n // 10)
    print(n % 10)
    return

# Chamada da  função
exibeVert (875)
```

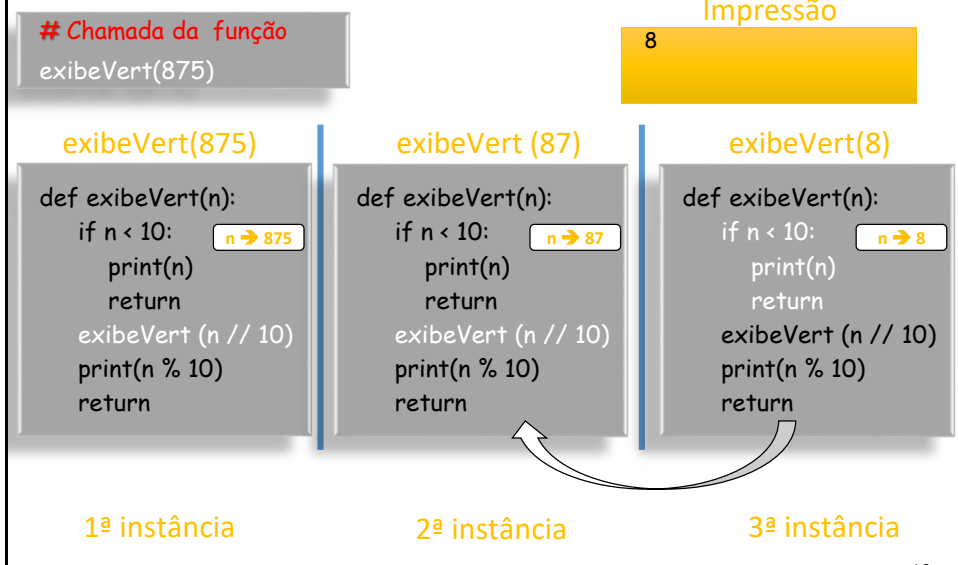
14

Simulação da execução



15

Simulação da execução



16

Simulação da execução

```
# Chamada da função
exibeVert(875)
```

Impressão

```
8
7
```

exibeVert(875)

```
def exibeVert(n):
    if n < 10:          n → 875
        print(n)
        return
    exibeVert (n // 10)
    print(n % 10)
    return
```

exibeVert (87)

```
def exibeVert(n):
    if n < 10:          n → 87
        print(n)
        return
    exibeVert (n // 10)
    print(n % 10)
    return
```

1ª instância

2ª instância

17

Simulação da execução

```
# Chamada da função
exibeVert(875)
```

Impressão

```
8
7
5
```

exibeVert(875)

```
def exibeVert(n):
    if n < 10:          n → 875
        print(n)
        return
    exibeVert (n // 10)
    print(n % 10)
    return
```

1ª instância

18

Simulação da execução

```
# Chamada da função
exibeVert(875)
```

Impressão

```
8
7
5
```

✓ Cada algarismo do número 875 foi impresso em uma linha