

Exercícios de Matemática Discreta e Programação

2011-2012

Folha 1

Programação

1. Escreva a função

```
def mod3(n):  
    return n % 3
```

e execute-a para diferentes argumentos inteiros. Qual a interpretação para o operador `%` em Python? Qual o comportamento da função para números negativos?

2. Defina a função **incrementaUmaUnidade**, que tenha por argumento x e devolva $x + 1$. Teste a sua função para $x = 3$, $x = 5$ e $x = 1.5$.
3. Defina a função **somaDe1AteN(n)**, que devolve $1 + 2 + \dots + n$ usando a fórmula $1 + 2 + \dots + n = \frac{n(n+1)}{2}$. Teste a função para vários valores de n .
4. Defina a função **inverso(x)**, que devolve $1/x$. Aplique a função para $x = 0$. Como é que o Python trata o domínio natural numa função?
5. Assumindo definida a função **duplica(x)**, que devolve $2x$, e a função **incrementaUmaUnidade(x)**, do exercício 2. Qual é o resultado de **duplica(incrementaUmaUnidade(6))**? Qual é o resultado de **incrementaUmaUnidade(duplica(6))**? Explique esses resultados.
6. Tente executar **incrementaUmaUnidade**, do exercício 2 com argumento `'123'`. Podemos adicionar um número a uma *string* em Python?
7. Tente executar **duplica**, do exercício 5 com argumento `'123'`. Qual o comportamento do operador `*` quando tem por operandos um inteiro e uma *string*?
8. No Python, se s é uma *string*, $s[0]$ identifica o primeiro carácter. Defina e teste uma função que quando aplicada a uma função devolva o seu primeiro carácter.
9. No Python, $[a, b, c, \dots, x]$ representa uma lista de objectos. Por exemplo $[1, 5, 2]$ representa a lista com três números: 1, 5 e 2. Qual é o comportamento da função do exercício quando tem por argumento uma lista?
10. Qual o resultado de aplicar as funções *sum*, *min* e *max* (funções pré-definidas) a uma lista de números?
11. Qual o resultado de executar $\min(\text{range}(n))$ e $\max(\text{range}(n))$, quando n é um inteiro positivo?
12. Reescreva a função *somaDe1AteN* do exercício 3 por forma a usar funções apresentadas nos exercícios 10 e 11.
13. Apresente o fluxograma de um algoritmo para calcular $s(n) = 1 + 2 + \dots + n$, sem recorrer à fórmula $s(n) = \frac{n(n+1)}{2}$. Apresente o algoritmo em pseudo-código.

14. Apresente o fluxograma de um algoritmo, em que usando apenas adições e diferenças, calcule o quociente e o resto da divisão de dois números. Apresente o algoritmo em pseudo-código.
15. Apresente o fluxograma de um algoritmo para calcular 3^n , onde n é um inteiro não negativo.
16. Apresente o fluxograma de um algoritmo, em que dado um inteiro n determine o maior k tal que $2^k \leq n$.
17. Assumindo que **palavra** é uma cadeia de letras minúsculas, que **palavra[i]** identifica a letra de ordem i em **palavra** (sendo **palavra[0]** a primeira letra) e que **len(palavra)** devolve o número de letras da cadeia. Apresente o fluxograma de um algoritmo que, dadas duas palavras, decida qual delas vem primeiro no dicionário.
18. Explique o resultado da execução de **2+-2** e **2++2**.
19. Experimente **2++++2**. Explique o resultado.
20. Experimente **2**3** e **2**4**. Descreva o comportamento do operador ******.
21. Experimente **"abc"+"def"** e **'abc'+ 'def'**. Descreva o comportamento do operador **+** quando aplicado a *strings*.
22. O operador ***** pode ser aplicado a *strings*? Experimente **3 * '12'** e explique o resultado.
23. Execute **9-8*2+6** e **(5-1)*(1+2)**3**. Qual a ordem de precedência dos operadores usados nas expressões?
24. Definindo a função:

```
def inverso(x):
    return 1/x
```

Qual o resultado de executar

```
>>> 1 + inverso(2*5)
```

Como é feita a avaliação quando um dos operadores é uma função? Como é feita a avaliação quando uma função é aplicada a uma expressão?

25. A função abaixo devolve o primeiro caracter duma *string*:

```
def primeiro(s):
    return s[0]
```

Adicione à função uma *string* de documentação. Execute:

```
>>> primeiro('Bom dia')
```

e

```
>>> primeiro.__doc__
```

26. Identifique os erros de sintaxe na definição da função abaixo:

```
def codigoErrado(x):  
    Return x**2 - 1
```

27. No Python, `s[-1]` identifica o último carácter duma *string* `s` (ou o último elemento duma lista `s`). Escreva uma função em que, de uma *string* `s` construa uma nova *string* contendo dois caracteres: o primeiro e o último carácter em `s`. Introduza um *string* de documentação no seu código.
28. Escreva uma função `triângulo(s)`, onde usa a função `print` uma única vez e que tenha por saída:

```
sssss  
sss  
s
```

onde `s` é uma *string* com um único carácter. Por exemplo, `triângulo(*)` deve devolver:

```
*****  
***  
*
```