

Aula 04- Comandos de Seleção

Introdução à Computação

Professor: Paulo Meirelles (paulormm@ime.usp.br)

Departamento de Ciência da Computação
Instituto de Matemática e Estatística



1. Divisão Inteira

Divisão Inteira em Python

- A divisão inteira em Python é feita pelo operador //

```
>>> 38 // 5
```

```
7
```

```
>>> 38.5 // 5.7
```

```
6.0
```

```
>>> 38.5/5.7
```

```
6.754385964912281
```

- O resto da divisão é dado pelo operador %

```
>>> 38 % 5
```

```
3
```

Extração de dígitos de Inteiro

```
>>> x = 6583
```

```
>>> x % 10
```

3

```
>>> x = x // 10
```

```
>>> x % 10
```

8

```
>>> x = x // 10
```

```
>>> x % 10
```

5

```
>>> x = x // 10
```

```
>>> x % 10
```

6

Usos do resto: divisibilidade

```
>>> x = 5
```

```
>>> x % 2 == 0 #Testa se x é par.
```

```
False
```

```
>>> x % 2 == 1 #Testa se x é ímpar.
```

```
True
```

```
>>> x = 6
```

```
>>> x % 2 == 0 #Testa se x é par.
```

```
True
```

```
>>> x % 2 == 1 #Testa se x é ímpar.
```

```
False
```

2. Algoritmo de Euclides

Algoritmo de Euclides

- Calcula o máximo divisor comum entre dois números inteiros

$$\text{mdc}(a,b)$$

- Método: restos sucessivos da divisão inteira
- É considerado o algoritmo não-trivial mais antigo, presente no livro *Elementos* de Euclides
- Vamos nos concentrar no caso

$$a \geq 0 \text{ e } b \geq 0$$

Princípios e Exemplos

- $\text{mdc}(a,b) = \text{mdc}(b, r)$, onde $r = a \% b$

Ou seja, o valor do mdc permanece o mesmo ao substituirmos a por r .

- $\text{mdc}(a,0) = a$. Essa é a condição de parada.

O método consiste em computar os restos sucessivos das divisões inteiras até encontrar o resto 0 (divisão exata).

- Ex1: 35, 15, **5** [= 35 % 15], 0 [= 15 % 5]. Ou seja

$$\text{mdc}(35,15) = \text{mdc}(15, 5) = \text{mdc}(5, 0) = 5$$

- Ex2: 108, 45, 18, **9**, 0. $\text{mdc}(108,45) = 9$

Implementação do Algoritmo de Euclides

- **Não** é necessário conhecer a demonstração dos princípios algébricos por trás do Algoritmo de Euclides para implementá-lo.
- Para a implementação, basta entender a mecânica dos **restos sucessivos**.
- Três variáveis: a , b e r
- Inicialização: ler $a, b \geq 0$
- A cada iteração, calculamos $r = a \% b$, trocamos os valores de a por b , e b por r .
- A condição de parada é $r == 0$
- Quando o laço pára, o mdc é o valor de a
- Cuidado: os valores lidos foram alterados!

3. Saída Formatada

```
>>> a = 2
>>> print("{:.2f}".format(a))
2.00
```

```
>>> b = 2.0
>>> print("{:.2f}".format(b))
2.00
```

```
>>> c = 2.00
>>> print("{:.2f}".format(c))
2.00
```

```
>>> d = 2.0000
>>> print("{:.2f}".format(d))
2.00
```

```
>>> e = 2.000000000
>>> print("{:.2f}".format(e))
2.00
```

```
>>> import math
```

```
>>> pi = math.pi
```

```
>>> e = math.e
```

```
>>> print("Pi = {:.4f} e Euler = {:.4f}".format(pi,e))
```

Pi = 3.1416 e Euler = 2.7183

Para as variáveis:

a = 13.22784

b = 1200.20004

c = 13227.84

d = 37.6

e = 0.02

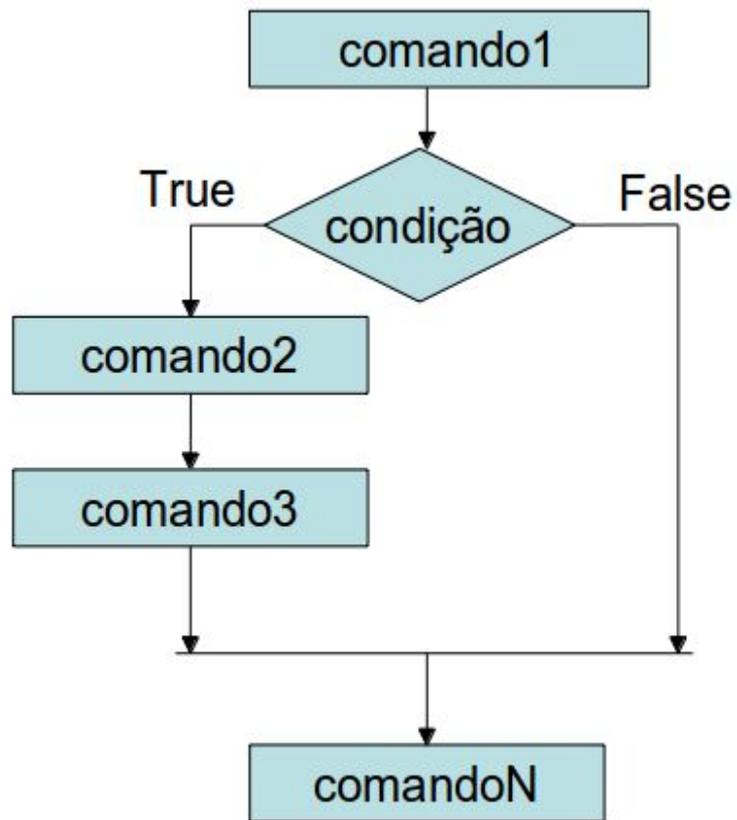
f = 127.0

faça um programa que exibe os valores com três casas de precisão após a vírgula, e na forma de uma tabela.

Solução:

```
1 print("|{:9.3f}|{:9.3f}|".format(a, b))
2 print("|{:9.3f}|{:9.3f}|".format(c, d))
3 print("|{:9.3f}|{:9.3f}|".format(e, f))
```

3. Estrutura Condicional Simples



comando1

if condição :

bloco de comandos.

comando2

comando3

:

comandoN

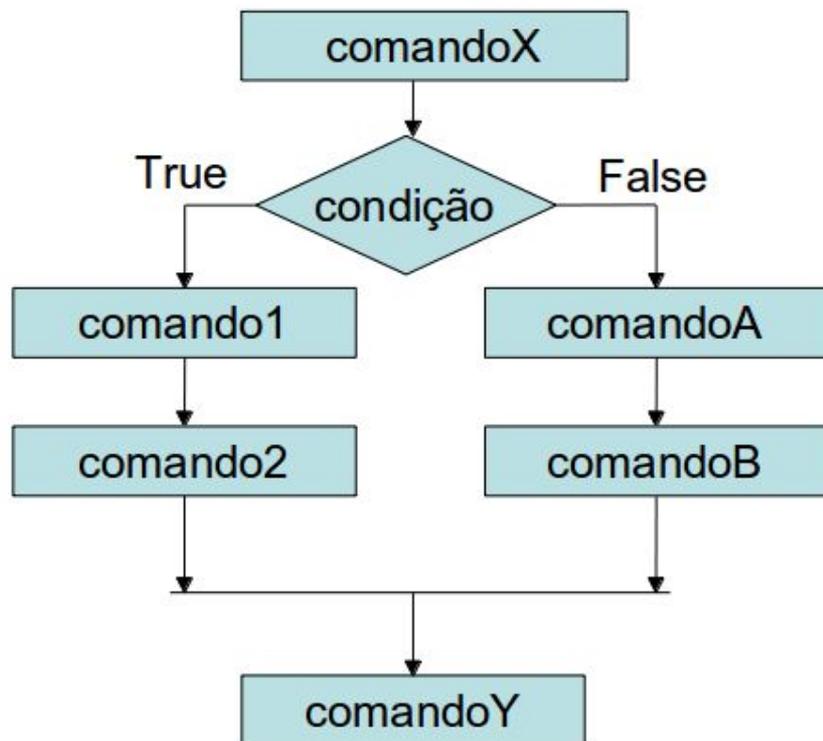
Contagem de Pares

Dados um número inteiro n , $n > 0$, e uma sequência com n números inteiros, determinar quantos números da sequência são pares. Por exemplo, para a sequência

6 -2 7 0 -5 8 4

o seu programa deve escrever o número 5 para o número de pares.

4. Estrutura Condicional Composta



comandoX

if condição :

bloco de comandos.

comando1

:

else :

bloco de comandos.

comandoA

comandoB

:

comandoY

Contagem de Pares e Ímpares

Dados um número inteiro n , $n > 0$, e uma sequência com n números inteiros, determinar quantos números da sequência são pares e quantos são ímpares.

Por exemplo, para a sequência

6 -2 7 0 -5 8 4

o seu programa deve escrever o número 5 para o número de pares e 2 para o de ímpares

Introdução à Computação

Professor: Paulo Meirelles
E-mail: paulormm@ime.usp.br