

# MC-102 — Aula 07

## Comandos Repetitivos

Instituto de Computação – Unicamp

25 de Agosto de 2015

# Roteiro

- 1 Variável Indicadora
  - Números Primos
  - Números em Ordem
- 2 Variável Contadora
  - Números Primos
- 3 Outros Exemplos
  - Maior Número
  - Números de Fibonacci
- 4 Exercícios

# Introdução

- Vimos quais são os comandos de repetição em Python.
- Veremos mais alguns exemplos de sua utilização.

# Variável Indicadora

- Um outro uso comum de laços é para a verificação se um determinado objeto, ou conjunto de objetos, satisfaz uma propriedade ou não.
- Um padrão que pode ser útil na resolução deste tipo de problema é o uso de uma **variável indicadora**.
  - ▶ Assumimos que o objeto satisfaz a propriedade (indicadora = True).
  - ▶ Com um laço verificamos se o objeto realmente satisfaz a propriedade. Se em alguma iteração descobrirmos que o objeto não satisfaz a propriedade, então fazemos (indicadora = False).

# Exemplo: Números Primos

## Problema

Determinar se um número  $n$  é primo ou não.

- Um número é primo se seus únicos divisores são 1 e ele mesmo.
- Dado um número  $n$  como detectar se este é ou não primo??
  - 1 Lê um número  $n$ .
  - 2 Testa se nenhum dos números entre 2 e  $(n - 1)$  divide  $n$ .
- Lembre-se que o operador  $\%$  retorna o resto da divisão.
- Portanto  $(a\%b)$  é zero se e somente se  $b$  divide  $a$ .

# Exemplo: Números Primos

Em Python:

```
n = int(input("Digite um número:"))
div = 2
eprimo=True
while div<=n-1 and eprimo :
    if n % div == 0:
        eprimo=False
        div = div + 1

if eprimo :
    print("É primo!!");
else:
    print("Não é primo!!");
```

# Exemplo: Números Primos

Com o uso de **break**:

```
n = int(input("Digite um número:"))
div = 2
eprimo=True
while div <= n-1 :
    if n % div == 0:
        eprimo=False
        break
    div = div + 1

if eprimo:
    print("É primo!!");
else:
    print("Não é primo!!");
```

## Exercício

- No exemplo dos números primos não precisamos testar todos os números entre  $2, \dots, (n - 1)$ , para verificar se dividem ou não  $n$ . Basta testarmos até  $n/2$ . Por que? Qual o maior divisor possível de  $n$ ?
- também não precisamos testar todos os números entre 2 e  $n/2$ , basta testar o 2, e de 2 em 2 entre 3 e  $n/2$
- Na verdade basta testarmos os números  $2, 3, \dots, \sqrt{n}$  de 2 em 2. Por que?



# Exemplo: Números em Ordem

## Problema

Fazer um programa que lê  $n$  números inteiros do teclado, e no final informa se os números lidos estão ou não em ordem crescente.

- Usaremos uma variável indicadora na resolução deste problema.

## Exemplo: Números em Ordem

- Um laço principal será responsável pela leitura dos números.
- Vamos usar duas variáveis, uma que guarda o número lido na iteração atual, e uma que guarda o número lido na iteração anterior.
- Os números estarão ordenados se a condição (anterior  $\leq$  atual) for válida durante a leitura de todos os números.

## Exemplo: Números em Ordem

```
n = int(input("Digite um número:"))
anterior = int(input())
i = 1 #leu um número
ordenado = True

while (i < n) and ordenado :
    atual = int(input())
    i = i + 1      #leu mais um número
    if atual < anterior:
        ordenado = False
    anterior = atual

if ordenado :
    print("Sequência está ordenada")
else:
    print("Sequência não está ordenada")
```

O programa acima pode não ler todos os números. Isso pode ser um problema (o usuário quer entrar  $n$  números mas o programa só le alguns antes de dar a resposta.) Modifique o programa para que ele sempre leia os  $n$  números, independentemente da resposta.

# Variável Contadora

- Considere ainda o uso de laços para a verificação se um determinado objeto, ou conjunto de objetos, satisfaz uma propriedade ou não.
- Um outro padrão que pode ser útil é o uso de uma **variável contadora**.
  - ▶ Esperamos que um objeto satisfaça  $x$  vezes uma sub-propriedade. Usamos um laço e uma variável que **conta** o número de vezes que o objeto tem a sub-propriedade satisfeita.
  - ▶ Ao terminar o laço, se contadora for igual à  $x$  então o objeto satisfaz a propriedade.

## Exemplo: Números Primos

- Um número  $n$  é primo se nenhum número de 2 até  $(n - 1)$  dividí-lo.
- Podemos usar uma variável que conta quantos números dividem  $n$ .
- Se o número de divisores for 0, então  $n$  é primo.

# Exemplo: Números Primos

```
n = int(input("Digite um número:"))

div = 2;
divisores=0;

while div <= n-1:
    if n%div == 0:
        divisores = divisores + 1
        div = div + 1

if divisores == 0:
    print("É primo!!")
else:
    print("Não é primo!!")
```

Modifique este programa para que os valores a serem testados (em `div`) sejam os discutidos acima (2, 3, ...  $\sqrt{n}$  de 2 em 2).

# Outros Exemplos

- O uso de variáveis **acumuladora**, **indicadora** e **contadora** são úteis em várias situações.
- Mas não existem fórmulas para a criação de soluções para problemas.
- Em outros problemas, o uso destes padrões pode aparecer em conjunto, ou nem mesmo aparecer como parte da solução.

# Maior Número

## Problema

Fazer um programa que lê  $n$  números do teclado e informa qual foi o maior número lido.

- O programa deve ter os seguintes passos:
  - 1 Lê um número  $n$ .
  - 2 Repete  $n$  vezes a leitura de um número determinando o maior.
- Como determinar o maior??



# Maior Número

```
n = int(input("Digite um número:"))

maior = int(input())
cont = 1 #leu um número
while cont<n:
    aux = int(input())
    if aux>maior:
        maior = aux
    cont = cont + 1 #leu mais um número

print("O maior número é: ", maior)
```

# Maior Número

## Usando o for

```
n = int(input("Digite um número:"))

maior = int(input())
for i in range(n-1):
    aux = int(input())
    if aux>maior:
        maior = aux

print("O maior número é: ", maior)
```

# Números de Fibonacci

- A série de Fibonacci é: 1, 1, 2, 3, 5, 8, 13, ...
- Ou seja o  $n$ -ésimo termo é a soma dos dois anteriores

$$F(n) = F(n - 1) + F(n - 2)$$

onde  $F(1) = 1$  e  $F(2) = 1$ .

## Problema

Fazer um programa que imprime os primeiros  $n$  números da série de fibonacci.

# Números de Fibonacci

```
n = int(input("Digite um número:"))

cont = 0 #conta quantidade de num. impressos
f_ant=0
f_atual=1
while cont < n : #enquanto não imprimiu n números
    print( f_atual, ", ", end="")
    cont = cont + 1 #imprimiu mais um número
    f_aux = f_atual
    f_atual = f_atual + f_ant
    f_ant = f_aux
```

# Exercício

- Considere o programa para determinar se uma sequência de  $n$  números digitados pelo usuário está ordenado ou não. Refaça o programa usando uma variável contadora ao invés de indicadora.

## Exercício

- Faça um programa que calcule o máximo divisor comum de dois números  $m, n$ . Você deve utilizar a seguinte regra do cálculo do mdc com  $m \geq n$

$$\text{mdc}(m, n) = m \text{ se } n = 0$$

$$\text{mdc}(m, n) = \text{mdc}(n, m \% n) \text{ se } n > 0$$