

BCM0505-22 – Processamento da Informação

Laços

Carla Negri Lintzmayer
carla.negri@ufabc.edu.br
<http://professor.ufabc.edu.br/~carla.negri/>

Outline

Introdução

Exemplos

Exercícios

Introdução

Laços

Às vezes precisamos repetir um conjunto de instruções várias vezes

- Em Python, podemos usar a estrutura de repetição `while`

While (Sintaxe)

```
1 while <expressão booleana>:  
2     instrução_1  
3     instrução_2  
4     ...  
5     instrução_k  
6 instrução_A
```

- O corpo do laço é definido pela indentação
- **instrução_A** é a primeira instrução fora do laço

While (Sintaxe)

```
1 while <expressão booleana>:  
2     instrução_1  
3     instrução_2  
4     ...  
5     instrução_k  
6 instrução_A
```

Funcionamento

1. Avalia **<expressão booleana>**. Se valor é **True**, executa o passo 2; caso contrário, executa passo 4
2. Executa as instruções do corpo
3. Volta para o passo 1
4. Executa a instrução **instrução_A** e continua a execução do programa

Exemplos

Números naturais

Um programa que imprime os n primeiros números naturais

```
1 n = int(input())
2 i = 1
3 while i <= n:
4     print(i)
5     i = i + 1
6 print("Fim")
```

Média

Um programa que lê n números fornecidos pelo usuário e imprime o valor médio

```
1  n = int(input())
2  i = 1
3  soma = 0
4  while i <= n:
5      val = float(input())
6      soma += val
7      i += 1
8  media = soma / n
9  print(f"Média: {media:.2f}")
```

Fatorial

Um programa que calcula $n!$

```
1 n = int(input())
2 i = n
3 fat = 1
4 while i > 1:
5     fat = fat * i
6     i = i - 1
7 print(f"{n}! = {fat}")
```

Fatorial

Um programa que calcula $n!$

```
1 def fatorial(n):
2     i = n
3     fat = 1
4     while i > 1:
5         fat = fat * i
6         i = i - 1
7     return fat
8
9 n = int(input())
10 resp = fatorial(n)
11 print(f"{n}! = {resp}")
```

- Encapsular a lógica do fatorial em uma função `fat(n)` poder ser útil no futuro!

Número Primo

Programa que determina se um dado número é primo

Número Primo

Programa que determina se um dado número é primo

```
1  n = int(input())
2  i = 2
3  primo = True
4  while i < n:
5      if n % i == 0:
6          primo = False
7          i = i + 1
8
9  if primo and n != 1:
10     print(f"{n} é primo")
11 else:
12     print(f"{n} não é primo")
```

Número Primo

Programa que determina se um dado número é primo

```
1  n = int(input())
2  i = 2
3  primo = True
4  while i < n:
5      if n % i == 0:
6          primo = False
7          i = i + 1
8
9  if primo and n != 1:
10     print(f"{n} é primo")
11 else:
12     print(f"{n} não é primo")
```

- Quais números estão sendo testados?

Número Primo

Programa que determina se um dado número é primo: uma primeira melhoria

Número Primo

Programa que determina se um dado número é primo: uma primeira melhoria

```
1  n = int(input())
2  i = 2
3  primo = True
4  while i < n and primo:
5      if n % i == 0:
6          primo = False
7          i = i + 1
8
9  if primo and n != 1:
10     print(f"{n} é primo")
11 else:
12     print(f"{n} não é primo")
```

Número Primo

Programa que determina se um dado número é primo: uma primeira melhoria

```
1  n = int(input())
2  i = 2
3  primo = True
4  while i < n and primo:
5      if n % i == 0:
6          primo = False
7          i = i + 1
8
9  if primo and n != 1:
10     print(f"{n} é primo")
11 else:
12     print(f"{n} não é primo")
```

- Qual o maior número de vezes que o laço `while` pode executar?

Número Primo

Programa que determina se um dado número é primo: uma segunda melhoria

- Se $n = pq$ e se $q \geq p$, então $n = pq \geq p^2 \Rightarrow \sqrt{n} \geq p$

Número Primo

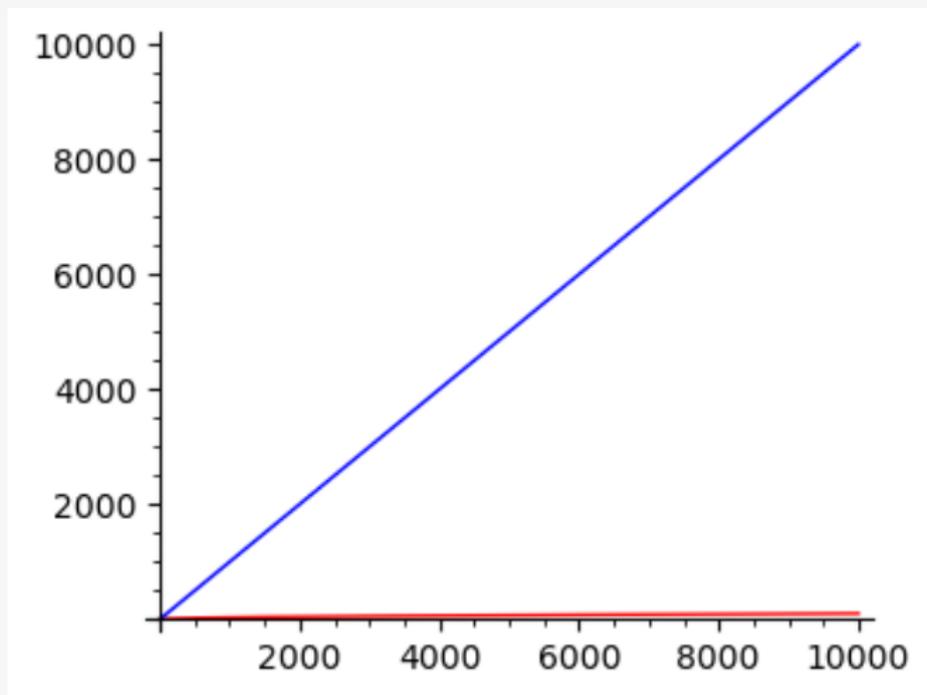
Programa que determina se um dado número é primo: uma segunda melhoria

- Se $n = pq$ e se $q \geq p$, então $n = pq \geq p^2 \Rightarrow \sqrt{n} \geq p$

```
1  from math import sqrt
2
3  n = int(input())
4  i = 2
5  primo = True
6  while i <= sqrt(n) and primo:
7      if n % i == 0:
8          primo = False
9          i = i + 1
10
11 if primo and n != 1:
12     print(f"{n} é primo")
13 else:
14     print(f"{n} não é primo")
```

- \sqrt{n} cresce muito mais devagar do que n ($\sqrt{n} \ll n$)

$$\sqrt{n} \ll n$$



Número Primo

Programa que determina se um dado número é primo: uma terceira melhoria

```
1  from math import sqrt
2  def eh_primo(n):
3      if n == 1:
4          return False
5      i = 2
6      while i <= sqrt(n):
7          if n % i == 0:
8              return False
9          i = i + 1
10     return True
11
12     n = int(input())
13     if eh_primo(n):
14         print(f"{n} é primo")
15     else:
16         print(f"{n} não é primo")
```

Ter uma função para computar se um número é primo pode vir a ser útil no futuro!

Exercícios

Exercícios

1. Dado um valor $n \in \mathbb{N}$, escreva um programa que imprima uma contagem regressiva. Por exemplo, se $n = 5$, então o seu programa poderia imprimir 5, 4, 3, 2, 1, FIM (um número por linha).
2. Escreva um programa que recebe n inteiros fornecidos pelo usuário e imprime o valor do maior número recebido.
3. Dado um valor $n \in \mathbb{N}$, escreva um programa que compute a soma dos *dígitos* de n .
4. Dado um valor $n \in \mathbb{N}$, escreva um programa que compute o n -ésimo número de Fibonacci F_n

$$F_n = \begin{cases} 1, & \text{se } n \leq 2 \\ F_{n-1} + F_{n-2}, & \text{se } n > 2 \end{cases}$$

n	1	2	3	4	5	6	7	8	9	10
F_n	1	1	2	3	5	8	13	21	34	55

5. Dado um valor $n \in \mathbb{N}$, escreva um programa que compute a soma de todos os números ímpares no intervalo $[0, n]$.
6. Dado um valor $n \in \mathbb{N}$, escreva um programa que imprima todos os números primos no intervalo $[0, n]$.