

7 Reunião 08: 24/SET/2020

7.1 Avisos

Provinha ~~10/7~~

*PO4 → Ver exercícios na
seção Exercícios da
página*

7.2 Aula passada

- execução condicional `if`
- execução alternativa `if-else`
- execução condicional em cadeia: `if-elif-else`
- operadores lógicos: `and` e `or` e `not`
- treino de raciocínio (sempre!) ← ☆



7.3 Comentários sobre a aula passada

- múltiplos de *i* e *j*: levamos a aula toda para resolver de 3 maneiras diferentes, com e sem operadores lógicos
- notar que nas soluções há uma variável responsável por contar o número de múltiplos exibidos.

7.4 Operadores e expressões lógicas

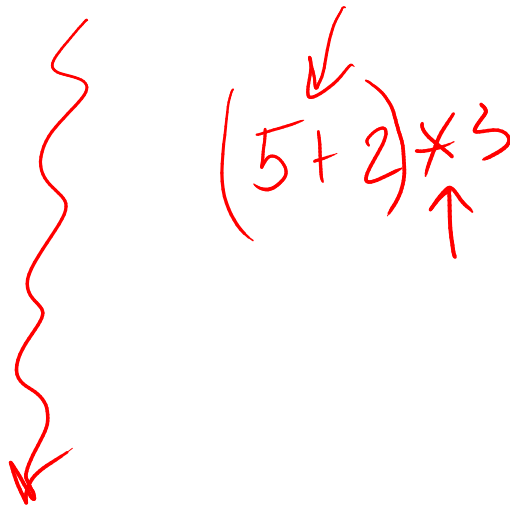
Existem três operadores lógicos: and, or e not ?

O resultados de condições é **True** ou **False**

Operadores **or** e **and**

7.5 Prioridades

- ()
- **
- * /
- + -
- < > <= >=
- != ==
- not**
- and**
- or**



Prioridade igual: da esquerda para a direita, exceto para o operador de exponenciação **

7.5.1 Exemplos

True and True é True

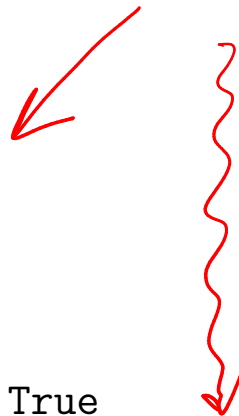
True and False é False

True or False é True

False or True é True

True or False and False é True

True and False or False é False



~~0~~ * = 2

7.6 Abreviaturas

i += 1 é equivalente a `i=i+1`

`+=` `*=` `/=` `--` `%=`

`i += j` # equivale a `i = i + j;`

`i += (j+k)*d` # equivale a `i = i + ((j+k)*d)`

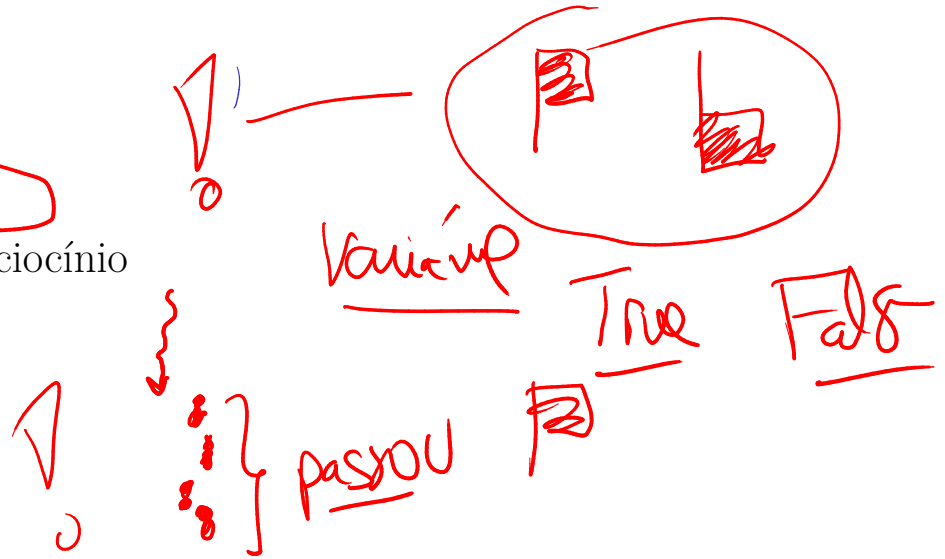
`i *= j+k` # equivale a `i = i*(j+k)`

`i = i + 1`
i += 1 → `a = a * 2`

`x operador= expressão` equivale a `x = x operador (expressão)`.

7.7 Hoje

- indicador de passagem
- mais treinamento de raciocínio



7.8 Exercício: sequência crescente

Dados um número inteiro n , $n > 0$, e uma sequência com n números inteiros, verificar se a sequência está em ordem crescente. Por exemplo, para a sequência

7 -6 0 12 15 37 101 201 a resp. é sim

7 -6 0 17 15 37 101 201 a resp. é não

7.8.1 Exemplos

Digite o tamanho da sequência: 5

Digite um número: -1

Digite um número: 10

Digite um número: 20

Digite um número: 36

Digite um número: 101

A sequência é crescente.

Digite o tamanho da sequência: 6

Digite um número: -10

Digite um número: 23

Digite um número: 37

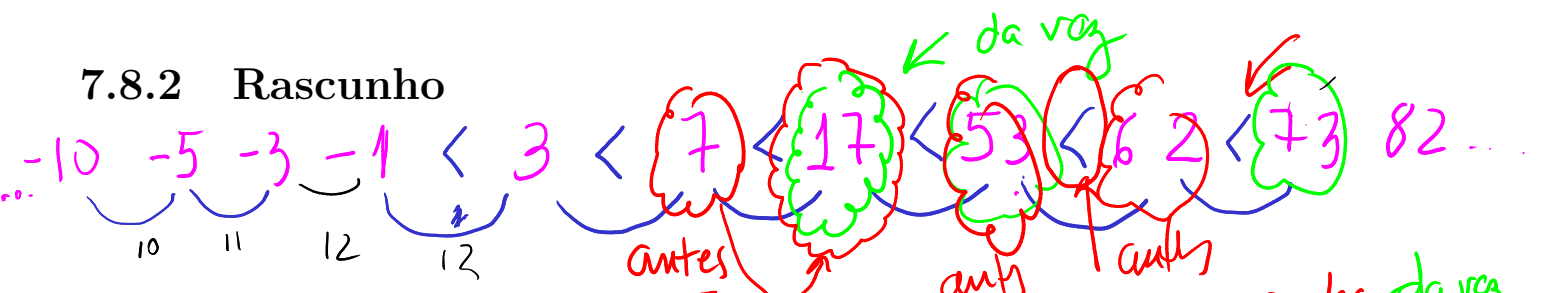
Digite um número: 35

Digite um número: 42

Digite um número: 67

A sequência não é crescente.

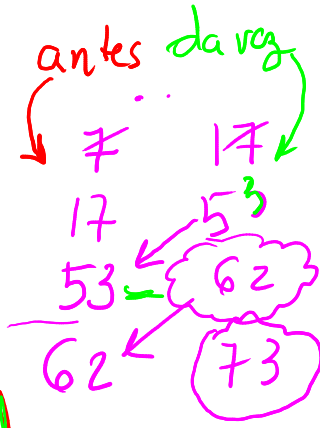
7.8.2 Rascunho



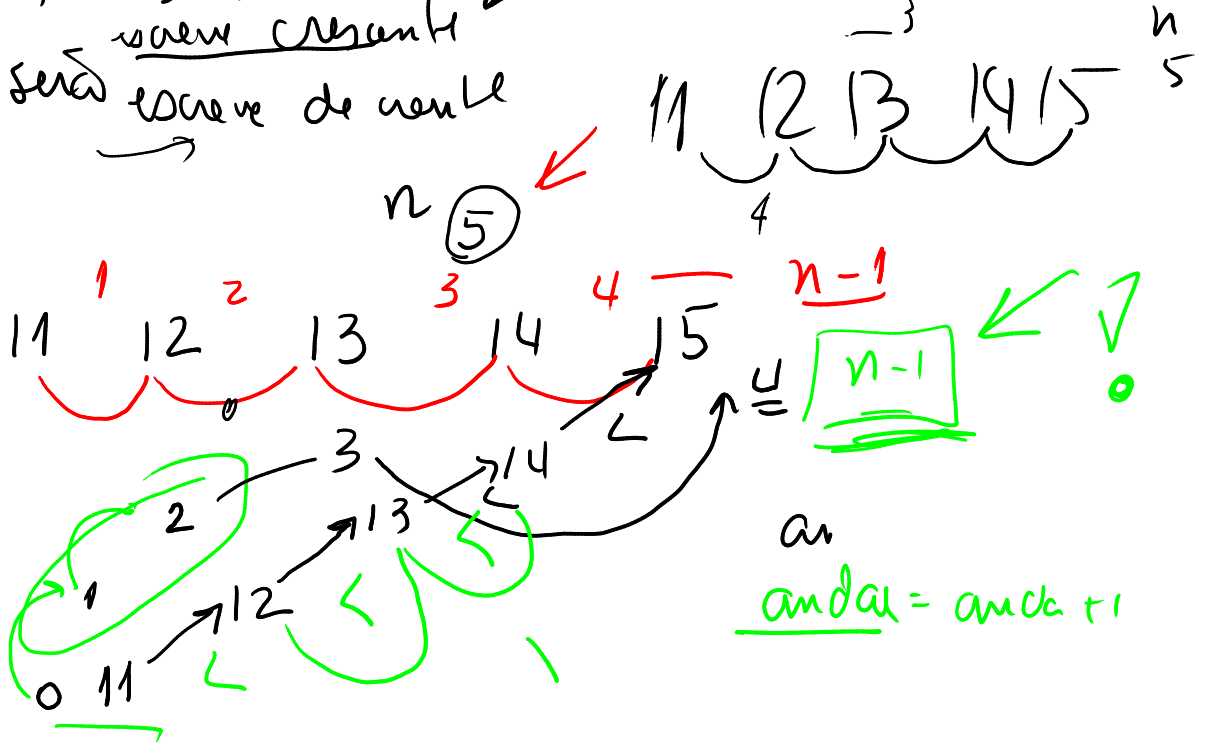
while
enquanto

```

da_vez ← ler(a)
if antes < da_vez:
    → x = x + 1
antes = da_vez # antes ← da_vez
    
```



→ $x == n-1$
será escrito de novo
será crescente



ou
anda = anda + 1

7.8.3 Solução

Uma solução

```
#leia o tamanho da sequencia
```

```
n = int(input("Digite o tamanho da sequência: "))
```

```
# indicador
```

```
crescente = True
```

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

```
i = 1
```

```
while i < n and crescente:
```

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

```
    if ant >= atual:
```

```
        crescente = False
```

```
    ant = atual
```

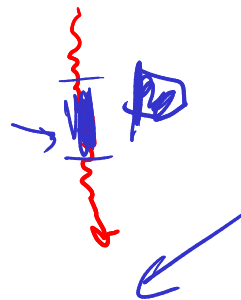
```
    i = i + 1
```

```
if crescente: # crescente = True  
    print("A sequencia é crescente. ")
```

```
else:
```

```
    print("A sequencia não é crescente. ")
```

Tópicos: indicador de passagem e mais expressões lógicas



7.9 Exercício: dígitos adjacentes iguais

Dado um número inteiro $n > 0$, verificar se este número contém dois dígitos adjacentes iguais.

Por exemplo, para

1234321 a resposta é não
1234556 a resposta é sim

7.9.1 Exemplos

Digite um número: 12345678901234567

12345678901234567 não possui dígitos adjacentes iguais.

Digite um número: 12345667890

12345667890 possui dois dígitos 6 adjacentes.

Digite um número: 112233445566

112233445566 possui dois dígitos 6 adjacentes.

Digite um número: 647365854

647365854 não possui dígitos adjacentes iguais.

Digite um número: 4543563534

4543563534 não possui dígitos adjacentes iguais.

7.9.3 Solução *é sempre uma solução*

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

```
anterior = n % 10
```

```
n = n // 10;
```

```
adj_iguais = False;
```

```
while n > 0 and not adj_iguais:
```

```
    atual = n % 10
```

```
    if atual == anterior:
```

```
        adj_iguais = True
```

```
    anterior = atual
```

```
    n = n // 10
```

```
print("-----")
```

```
if adj_iguais:
```

```
    print(n_salvo, "possui dois dígitos", atual, "adjacentes.")
```

```
else:
```

```
    print(n_salvo, "não possui dígitos adjacentes iguais.")
```

Tópicos: mais indicador de passagem