

10 Reunião 10: 01/OUT/2020

10.1 Aulas passadas

- treino de raciocínio (sempre!)

EP04 → Avaliador
discos

• Haik

13h à 14h
2^{as}

Monitoria



- indicadores de passagem
- execuções condicionais, alternativas e em cadeia: `if`, `if-else`, `if-elif-else`
- operadores lógicos: `and` e `or` e `not`
- abreviaturas: `i += 1` é o mesmo que `i = i + 1`; `pot2 *= 2` é o mesmo que `pot2 = pot2 * 2`



Figure 1: Fonte: depositphotos.com

10.2 Comentários sobre a aula passada

- tiro a alvo: fizemos uma solução passo a passo, quebrando a região em não muitos pedaços.



Figure 2: Fonte: yogajournal.com

10.3 Hoje

- mais indicadores de passagem ainda
- repetições encaixadas: `while ... dentro de while ... dentro de ...`

10.4 Exercício: números primos → saída

Dado um número inteiro n , $n > 0$, verificar se n é primo.

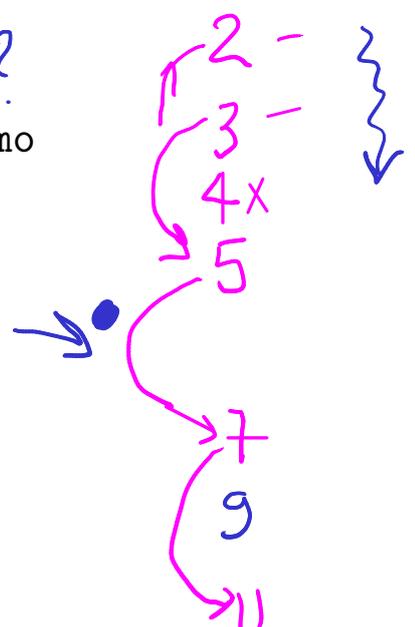
Um número inteiro maior que 1 é **primo** se seus únicos divisores positivos são 1 e ele mesmo.

Por exemplo,

- 1 não é primo
- 2 é primo
- 27644437 é primo
- 2147483647 é primo
- 35742549198872617291353508656626642567 é primo

$$2 \% 2 == 0$$

$$\begin{array}{r} 2 \mid 2 \\ 0 \mid 1 \end{array} \quad \begin{array}{r} 2 \mid 1 \\ 0 \mid 2 \end{array}$$



10.4.1 Exemplos

Digite um número: 2

2 é primo

Digite um número: 5

5 é primo

Digite um número: 6

6 não é primo: 6 % 2 = 0

120

Digite um número: 117

117 não é primo: 117 % 3 = 0

PAVSA

Conjectura: Todo número ímpar maior que 1 é primo. X

Gauss
Fermat
Hilbert

Matemático: 3[✓], 5[✓], 7[✓], 9[✗] (3)

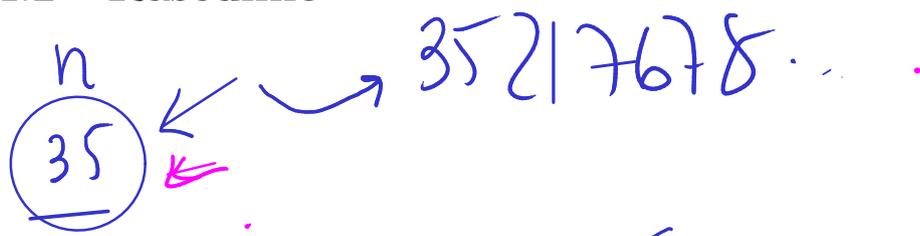
Físico: 3[✓], 5[✓], 7[✓], 9[✗], 11[✓] → Coelho

Engenheiro: 3[✓], 5[✓], 7[✓], 9[✓], 11[✓] | Palpate

Computólogo: 3[✓], 5[✓], 7[✓], 9[✓], 9[✓], 9[✓], 9[✓], 9[✓], ...

- Matemático : 3, 5, 7, 9 falso
- Físico : 3, 5, 7, 9... erro experimental, 11,...
- Engenheiro : 3, 5, 7, 9, 11, 13,
- Computólogo: 3, 5, 7, 9, 9, 9, 9, 9, ...

10.4.2 Rascunho



$e\text{-primo} = \text{True}$

```
if  $n \% 2 == 0$ :  
     $e\text{-primo} \leftarrow \text{False}$ 
```

```
if  $n \% 3 == 0$ :  
     $e\text{-primo} \leftarrow \text{False}$ 
```

```
if  $n \% 4 == 0$ :  
     $e\text{-primo} \leftarrow \text{False}$ 
```



10.4.3 Solução

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

```
div = 2
```

```
e_primo = True
```

```
while div*div <= n and e_primo: # == True:
```

```
    if n%div == 0:
```

```
        e_primo = False
```

```
    else:
```

```
        div = div + 1
```

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

```
if n == 1:
```

```
    print("1 não é primo.")
```

```
elif e_primo:
```

```
    print(f"{n} é primo.")
```

```
else:
```

```
    print(f"{n} não é primo: {n} % {div} = {n%div}")
```

O mesmo que $div \leq \sqrt{n}$
se n tem um divisor
então n tem um divisor $\leq \sqrt{n}$

$$a \times b = n$$

$\sqrt{a} \quad \sqrt{b} \quad \sqrt{n}$

10.5 Exercício: fatores primos



Dado um número inteiro positivo n determine a sua decomposição em fatores primos calculando também a multiplicidade de cada fator.

10.5.1 Exemplos

$$120 = 2^3 \times 3^1 \times 5^1 \times 1^1$$

Decomponho um inteiro positivo em fatores primos.

Digite n: 120

Decomposição de 120 em fatores primos:

fator 2 tem multiplicidade 3
fator 3 tem multiplicidade 1
fator 5 tem multiplicidade 1

fim

Decomponho um inteiro positivo em fatores primos.

Digite n: 4345

Decomposição de 4345 em fatores primos:

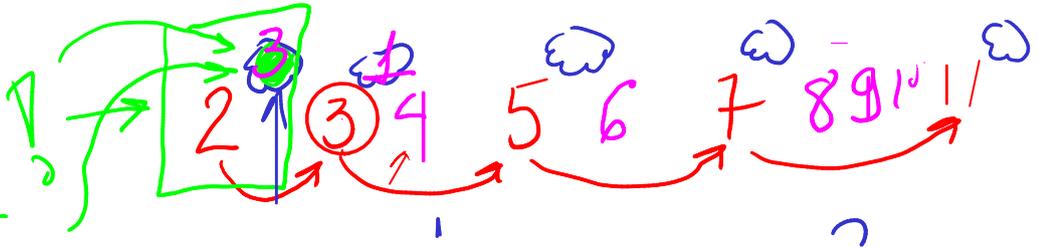
fator 5 tem multiplicidade 1
fator 11 tem multiplicidade 1
fator 79 tem multiplicidade 1

fim

$$4345 = 5^1 \times 11^1 \times 79^1$$

10.5.2 Rascunhos

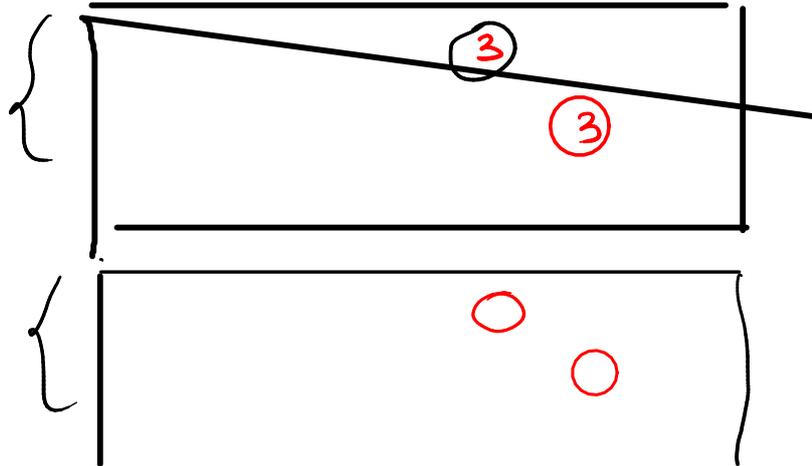
n	20	2	✓
	60	2	✓
	30	2	✓
	15	2x	
	15	3	✓
	5	3x	
	5	4x	
	5	5	✓
	1		



```

while i <= 8
{
    while n % div != 0
    {
        n = n // div
        mult = mult + 1
    }
    div += 1
}

```



10.5.3 Solução

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

```
print("Decomposição de", n, "em fatores primos:")
```

```
num = n # copia para salvar n
```

```
fator = 2
```

```
while num > 1:
```

```
    mult = 0
```

```
    while num % fator == 0:
```

```
        mult += 1
```

```
        num = num // fator
```

```
    if mult != 0:
```

```
        print("    fator", fator, "tem multiplicidade ", mult)
```

```
    fator += 1
```

```
print('fim')
```



Ideias!

Ideias