

# MC102 - Algoritmos e Programação de Computadores

MC102    Horários    Plano de desenvolvimento    Plano de aulas

Oferecimentos anteriores

**10 + 12 = 22**

**10 % -3 = ?? Aritmética com inteiros**

---

Nesta tarefa, daremos continuidade à familiarização com o interpretador Python e com a ferramenta de submissão SuSy. Você deverá ler valores na entrada e produzir as saídas esperadas. Além disso, verificaremos como funcionam as operações de divisão inteira e módulo em Python.

## Testes com a shell

---

Antes de escrever seu programa, vamos fazer alguns testes com a linha de comando. Abra um terminal e o programa `python3`:

```
$ python3
Python 3.7.1
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Atribua valores inteiros para as variáveis `a` e `b`:

```
>>> a = 7
>>> b = 3
```

**Soma, subtração, multiplicação.** Verifique o valor das operações básicas:

```
>>> a + b
10
>>> a - b
4
```

```
>>> a * b
21
```

**Divisão inteira.** Utiliza o operador `//` e irá retornar como quociente o valor  $q = \lfloor a/b \rfloor$ . Observe:

```
>>> a // b
2
```

Verifique o comportamento desta operação quando alteramos os sinais dos operandos. Lembre-se que a função piso é o arredondamento de um número para baixo e, portanto,  $\lfloor 2.33 \rfloor = 2$  e  $\lfloor -2.33 \rfloor = -3$ . Para facilitar a conferência dos valores, vamos imprimir as equações:

```
>>> print(a, "//", b, "=", a // b)
7 // 3 = 2
```

```
>>> print(-a, "//", b, "=", -a // b)
-7 // 3 = -3
```

```
>>> print(a, "//", -b, "=", a // -b)
7 // -3 = -3
```

```
>>> print(-a, "//", -b, "=", -a // -b)
-7 // -3 = 2
```

**Módulo** O operador `%` retorna o resto  $r$  de uma divisão inteira, tal que  $r = a - \lfloor a/b \rfloor * b$ . Observe a variação do resultado com a variação dos sinais dos operadores.

```
>>> print(a, "%", b, "=", a % b)
7 % 3 = 1
```

```
>>> print(-a, "%", b, "=", -a % b)
-7 % 3 = 2
```

```
>>> print(a, "%", -b, "=", a % -b)
7 % -3 = -2
```

```
>>> print(-a, "%", -b, "=", -a % -b)
-7 % -3 = -1
```

**Divisão real.** A divisão em Python com resultados reais tem como resultado um `float`. Este assunto será abordado em outra tarefa.

**Divisão por zero.** A divisão por zero não tem valor definido (leia [Wikipedia: Divisão por zero](#) ou veja o vídeo [Why can't you divide by zero? - TED-Ed](#)). Em Python, uma tentativa de dividir por zero ou tentar calcular o resto da divisão por zero levará à interrupção do programa devido a uma exceção.

```
>>> a // 0
Traceback (most recent call last):
  File "", line 1, in
ZeroDivisionError: integer division or modulo by zero
```

```
>>> a % 0
Traceback (most recent call last):
  File "", line 1, in
ZeroDivisionError: integer division or modulo by zero
>>>
```

## Descrição da entrada

---

A entrada do seu programa será composta por duas linhas, cada uma contendo um número inteiro. Observe o exemplo abaixo:

```
10
7
```

Dica: você poderá ler este par de valores com os comandos:

```
a = int(input())
b = int(input())
```

## Descrição da saída

---

A saída deverá reproduzir o valor lido das variáveis `a` e `b` e apresentar o conjunto de equações com as operações descritas nesta tarefa da seguinte forma:

```
a = 10
b = 7
a + b = 17
a - b = 3
a * b = 70
a // b = 1
a % b = 3
```

## Testes com o SuSy

---

No SuSy, para cada tarefa, criamos um conjunto de testes com arquivos de entrada `arq<i>.in` e para cada um deles temos uma saída esperada `arq<i>.res`. Para esta tarefa, todos os testes estão abertos e listados na tabela abaixo. Todos os segundos operandos são diferentes de zero.

Entrada		Saída	
	10	a = 10	
	5	b = 5	
arq1.in		a + b = 15	arq1.res
		a - b = 5	
		a * b = 50	
		a // b = 2	
		a % b = 0	

arq2.in	0 3	a = 0 b = 3 a + b = 3 a - b = -3 a * b = 0 a // b = 0 a % b = 0	arq2.res
arq3.in	-10 3	a = -10 b = 3 a + b = -7 a - b = -13 a * b = -30 a // b = -4 a % b = 2	arq3.res
arq4.in	10 -3	a = 10 b = -3 a + b = 7 a - b = 13 a * b = -30 a // b = -4 a % b = -2	arq4.res
arq5.in	-10 -3	a = -10 b = -3 a + b = -13 a - b = -7 a * b = 30 a // b = 3 a % b = -1	arq5.res

## Orientações para submissão

---

Veja [aqui](#) a página de submissão da tarefa. O arquivo a ser submetido deve se chamar `lab01.py`. No link [Arquivos auxiliares](#) há um arquivo [aux01.zip](#) que contém todos os arquivos de testes abertos e seus respectivos resultados compactados.

Utilize o sistema SuSy com o mesmo login e senha que você utiliza para fazer acesso ao sistema da DAC. Se você não estiver inscrito corretamente, envie email para [islene@ic.unicamp.br](mailto:islene@ic.unicamp.br).

O limite máximo será de 15 submissões. Serão considerados os resultados da última submissão.

O peso desta tarefa é 1.

O prazo final para submissão é 03/05/2020.

---