

## Exercícios Resolvidos da Aula 0: Fundamentos da programação estruturada

1. **Divisor:** Dados dois inteiros positivos  $x$  e  $y$ , verifique se  $x$  é divisor de  $y$ . O retorno deve ser booleano (verdadeiro ou falso).

```
1 def divisor(x, y):
2     if y%x == 0:
3         return True
4     else: return False
```

2. **Divisores:** Dado um número  $k$ , imprima todos os divisores de  $k$ .

```
1 from divisor import divisor
2
3 def divisores(k):
4     x = 1
5     while x <= k:
6         if divisor(x, k):
7             print(x, end=" ")
8         x = x+1
9     print()
```

3. **Máximo Divisor Comum (MDC):** Dados dois números  $m$  e  $n$ , imprima o máximo divisor comum entre  $m$  e  $n$ .

```
1 from divisor import divisor
2
3 def maior(m, n):
4     if m > n:
5         return m
6     else:
7         return n
8
9 def mdc(m, n):
10     i = 1
11     maximo = 1
12     while i <= maior(m, n):
13         if divisor(i, m) and divisor(i, n):
14             maximo = i
15         i = i+1
16     print("MDC({}, {}) = {}".format(m, n, maximo))
```

4. **Primo:** Dado um número  $x$ , verifique se ele é primo.

```

1 from divisor import divisor
2
3 def primo(x):
4     i = 2
5     divisores = 0
6     while i < x:
7         if divisor(i, x):
8             divisores = divisores + 1
9             i = i+1
10    # x eh primo se ele tem zero divisores alem de 1 e ele mesmo
11    return divisores == 0

```

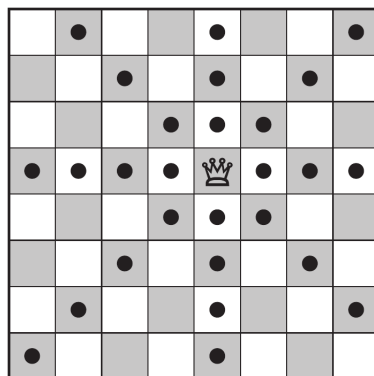
5. **Primos:** Dado um número  $k$ , imprima todos os números primos até  $k$ .

```

1 from divisor import divisor
2 from primo import primo
3
4 def primos(k):
5     i = 2
6     while i <= k:
7         if primo(i):
8             print(i, end=" ")
9             i = i+1
10    print()

```

6. **Dama:** [Maratona de Programação 2008] O jogo de xadrez possui várias peças com movimentos curiosos. Uma delas é a dama, que pode se mover qualquer quantidade de casas na mesma linha, na mesma coluna, ou em uma das duas diagonais, conforme exemplifica a Figura 1.



**Figura 1:** Movimentos possíveis da dama em um tabuleiro de xadrez.

Dada duas posições  $(x, y)$  e  $(m, n)$  em um tabuleiro de xadrez vazio (ou seja, um tabuleiro  $8 \times 8$ , com 64 casas), calcule e imprima a quantidade mínima de movimentos que a dama precisa fazer para ir da posição  $(x, y)$  para a posição  $(m, n)$ . Exemplos:

<i>Entrada</i>		<i>Saída</i>
$(x, y)$	$(m, n)$	
(4, 4)	(6, 2)	1
(3, 5)	(3, 5)	0
(5, 5)	(4, 3)	2

```

1 def abs(x):
2     if x < 0: return -x
3     else: return x
4
5 def mesmaPos(x1, y1, x2, y2):
6     return x1 == x2 and y1 == y2
7
8 def alinhados(x1, y1, x2, y2):
9     return x1 == x2 or y1 == y2 or (abs(x1-x2) == abs(y1-y2))
10
11 def dama(x, y, m, n):
12     if (mesmaPos (x, y, m, n)): print(0)
13     elif (alinhados (x, y, m, n)): print(1)
14     else: print(2)

```

7. **Acerola:** [Maratona de Programação 2008] Natural das Antilhas, a acerola (Malpighia glabra Linn, também conhecida como cereja das Antilhas) já era apreciada pelos nativos das Américas há muitos séculos. Mas o grande interesse por essa fruta surgiu na década de 1940, quando cientistas porto-riquenhos descobriram que a acerola contém grande quantidade de ácido ascórbico (vitamina C). A acerola apresenta, em uma mesma quantidade de polpa, até 100 vezes mais vitamina C do que a laranja e o limão, 20 vezes mais do que a goiaba e 10 vezes mais do que o caju e a amora.

Um grupo de amigos está visitando o Sítio do Picapau Amarelo, renomado produtor de acerola. Com a permissão de Dona Benta, dona do sítio, colheram uma boa quantidade de frutas, e pretendem agora fazer suco de acerola, que será dividido igualmente entre os amigos durante o lanche da tarde.

Conhecendo o número de amigos, a quantidade de frutas colhidas, e sabendo que cada unidade da fruta é suficiente para produzir 50 ml de suco, escreva uma função que receba como parâmetros o número  $N$  de amigos e a quantidade  $F$  de frutas

colhidas, e imprima com precisão de duas casas decimais qual o volume, em litros, que cada amigo poderá tomar. Exemplos:

<i>Entrada</i>		<i>Saída</i>
<i>N</i>	<i>F</i>	
1	1	0.05
5	431	4.31
101	330	0.16

```

1 def acerola(N, F):
2     q = F*0.05
3     print("%.2f"%(q/N))

```

8. **Alarme Despertador:** [Maratona de Programação 2009] Daniela é enfermeira em um grande hospital e tem os horários de trabalho muito variáveis. Para piorar, ela tem sono pesado, e uma grande dificuldade para acordar com relógios despertadores. Recentemente ela ganhou de presente um relógio digital, com alarme com vários tons, e tem esperança que isso resolva o seu problema. No entanto, ela anda muito cansada e quer aproveitar cada momento de descanso. Por isso, carrega seu relógio digital despertador para todos os lugares, e sempre que tem um tempo de descanso procura dormir, programando o alarme despertador para a hora em que tem que acordar.

No entanto, com tanta ansiedade para dormir, acaba tendo dificuldades para adormecer e aproveitar o descanso. Um problema que a tem atormentado na hora de dormir é saber quantos minutos ela teria de sono se adormecesse imediatamente e acordasse somente quando o despertador tocasse. Mas ela realmente não é muito boa com números, e pediu sua ajuda para escrever uma função que, dada a hora corrente e a hora do alarme, determine o número de minutos que ela poderia dormir. Exemplos:

<i>Entrada</i>				<i>Saída</i>
<i>Hora atual</i>	<i>Minuto atual</i>	<i>Hora alarme</i>	<i>Minuto alarme</i>	
1	5	3	5	120
23	59	0	34	35
21	33	21	10	1417

```

1 def alarme(hora_atual, minuto_atual, hora_alarme, minuto_alarme):
2     t1 = hora_atual*60 + minuto_atual
3     t2 = hora_alarme*60 + minuto_alarme
4
5     if (t1 < t2):

```

```
6         # Dorme e acorda no mesmo dia:  
7         print(t2-t1)  
8     else:  
9         # Dorme e acorda em dias diferentes  
10        print (24*60 + t2-t1)
```