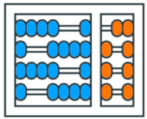


Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

MC102 - Aula 13

Exemplos: Funções

Algoritmos e Programação de Computadores

Turmas

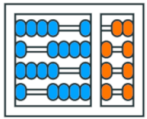
OVXZ

Prof. Lise R. R. Navarrete

lrommel@ic.unicamp.br

Quinta-feira, 05 de maio de 2022

19:00h - 21:00h (CB06)



Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

MC102 – Algoritmos e Programação de Computadores

Turmas

OVXZ

https://ic.unicamp.br/~mc102/

Site da Coordenação de MC102

Aulas teóricas:

Terça-feira, 21:00h - 23:00h (CB06)

Quinta-feira, 19:00h - 21:00h (CB06)

Conteúdo

- Exemplo 1
- Exemplo 2
- Exemplo 3
- Exemplo 4

Exemplo 1

Exemplo 1

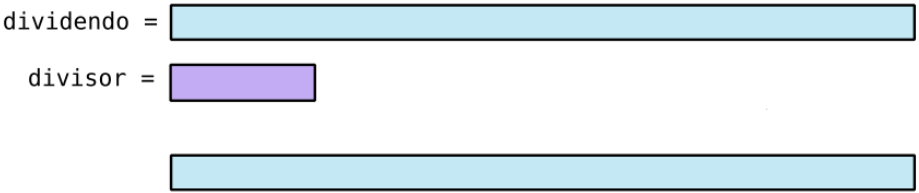
Crie uma função, tal que, dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.

Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.

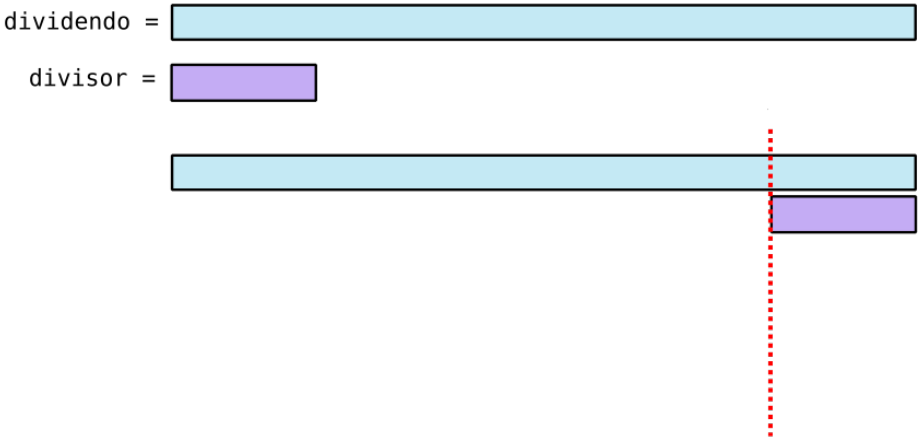
dividendo = 

divisor = 

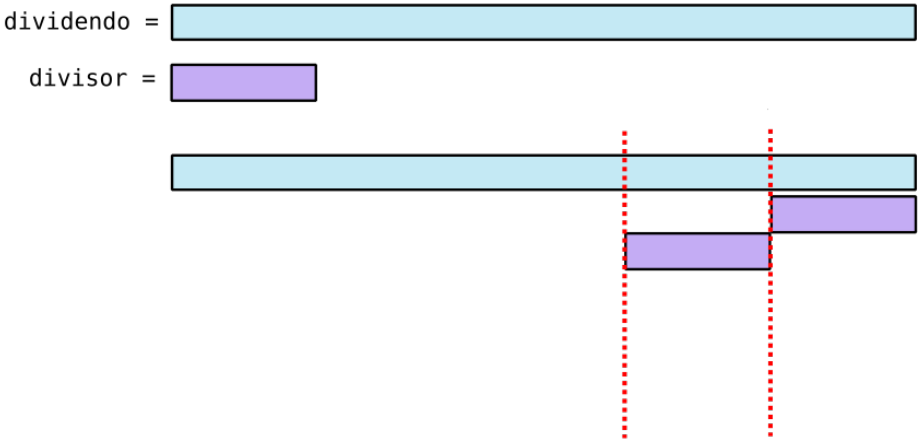
Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



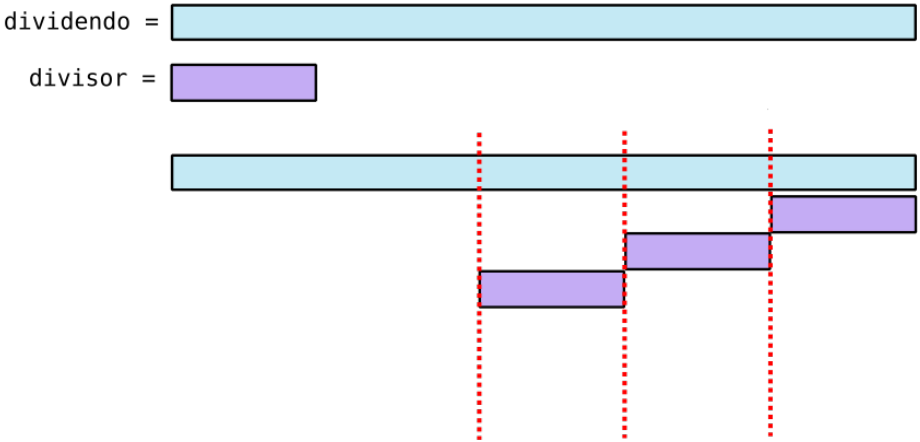
Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



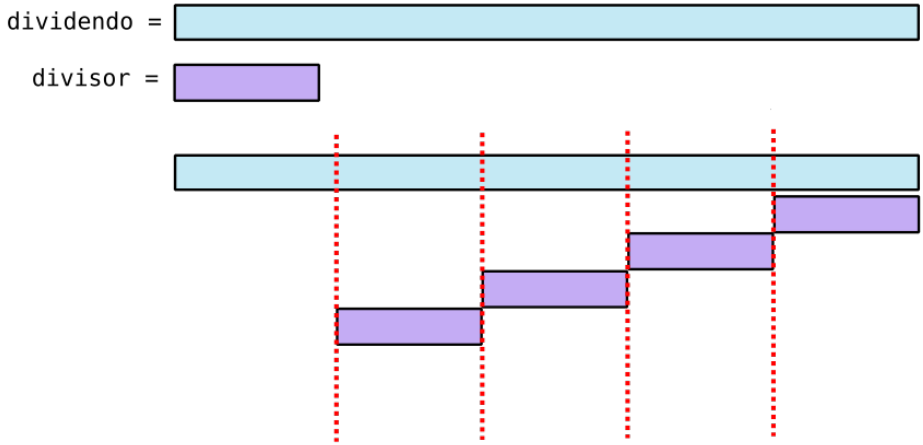
Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



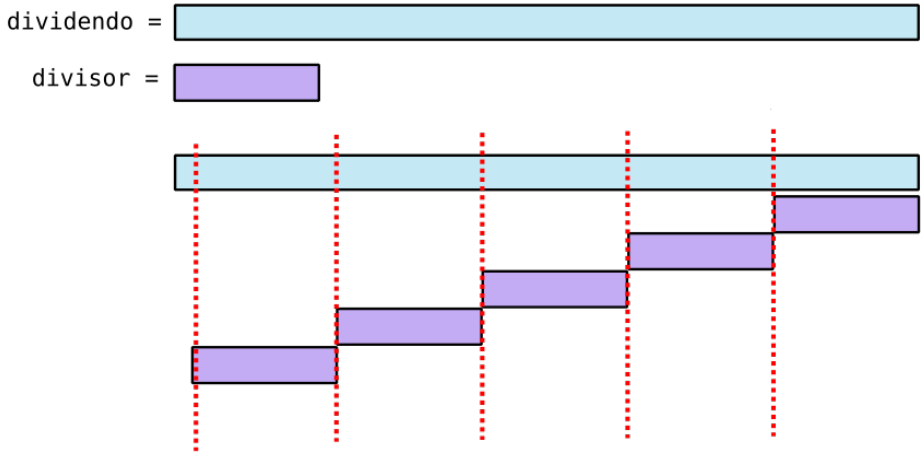
Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



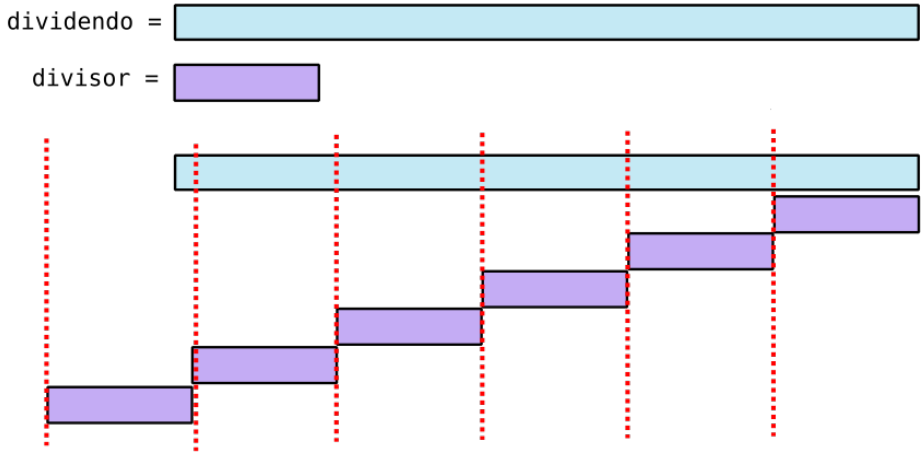
Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



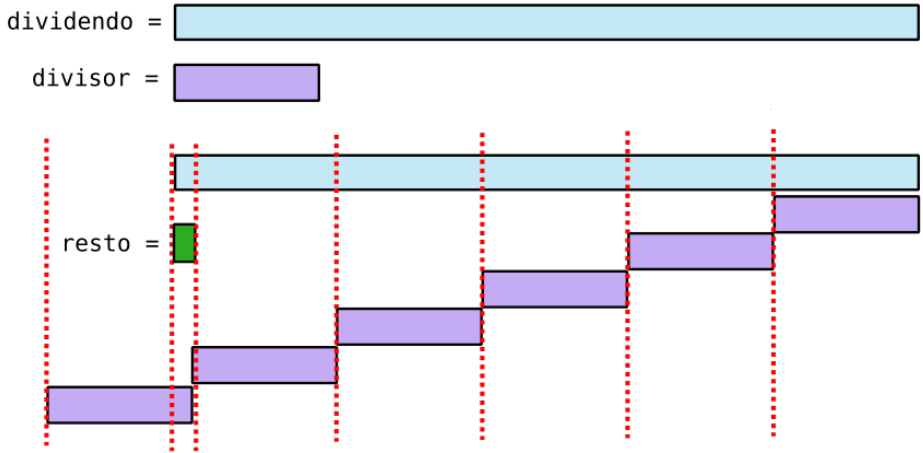
Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



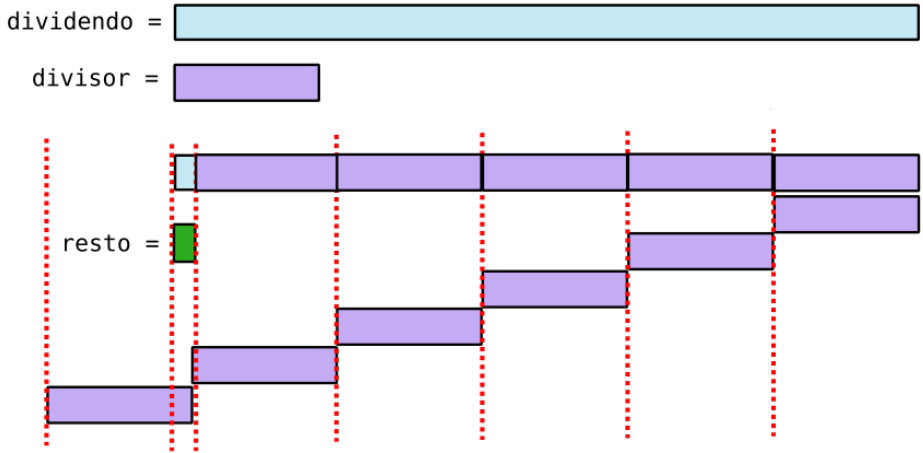
Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.



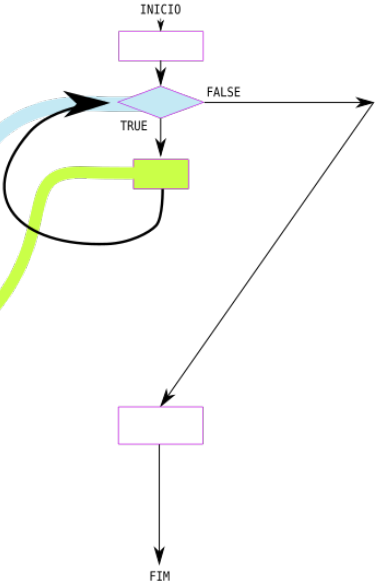
Dados dois números inteiros positivos, calcule o quociente e o resto da divisão inteira entre os dois, usando apenas somas e subtrações.

```
dividendo = int(input("Entre com o dividendo: "))
divisor = int(input("Entre com o divisor: "))

quociente = 0

while dividendo >= divisor:
  dividendo = dividendo - divisor
  quociente = quociente + 1

print("Quociente:", quociente)
print("Resto:", dividendo)
```



```python
def operar(a,b):
    (a,b) = min(a,b), max(a,b)

    return (a,b)

R = operar(54,3)

print(operar(54,3))
print(R)
print(R[0],R[1])

```

```
$ python3 exe1000.py
(3, 54)
(3, 54)
3 54
$
```

https://shell.cloud.google.com/

```python
def operar(D,d):
    q = 0       #contador

    while (D > d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador

    return (q,D)

print(operar(54,3))
```

```
$ python3 exe1000.py
(17, 3)
$
```

https://shell.cloud.google.com/

```python
def operar(D,d):
    q = 0       #contador

    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador

    return (q,D)

print(operar(54,3))
```

```
$ python3 exe1000.py
(18, 0)
$
```

https://shell.cloud.google.com/

```python
def operar(D,d):
    q = 0       #contador

    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador

    return (q,D)

print(operar(54.7,3))
```

```
$ python3 exe1000.py
(18, 0.7000000000000028)
$
```

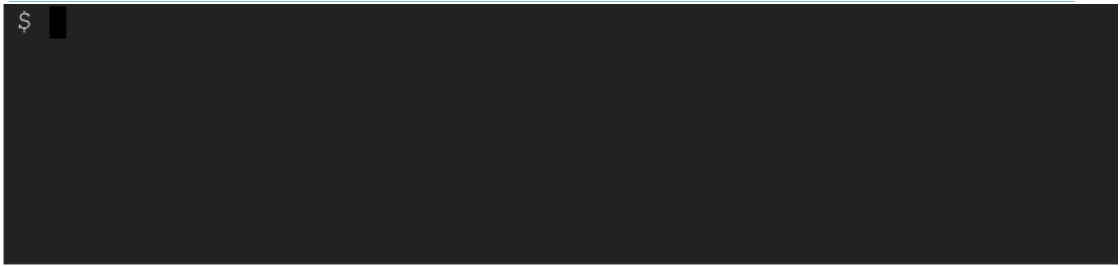
https://shell.cloud.google.com/

```python
import sys
x1,x2 = int(sys.argv[1]), int(sys.argv[2])

def operar(D,d):
    q = 0       #contador
    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador
    return (q,D)

print(operar(x1,x2))

```



https://shell.cloud.google.com/

```python
import sys
x1,x2 = float(sys.argv[1]), float(sys.argv[2])

def operar(D,d):
    q = 0       #contador
    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador
    return (q,D)

print(operar(x1,x2))

```

```
$ python3 exe1000.py 54 3
(18, 0.0)
$ python3 exe1000.py 54.7 3
(18, 0.7000000000000028)
$
```

https://shell.cloud.google.com/

```python
import sys
x1,x2 = int(sys.argv[1]), int(sys.argv[2])

def operar(D,d):
    q = 0       #contador
    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador
    return (q,D)

print(operar(x1,x2))
```

```
$ python3 exe1000.py 201 20
(10, 1)
$
```

https://shell.cloud.google.com/

```python
import sys
x1,x2 = int(sys.argv[1]), int(sys.argv[2])

def operar(D,d):
    q = 0       #contador
    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador
    return (q,D)

print(operar(x1,x2))
```

```
$ python3 exe1000.py 201 20
(10, 1)
$ python3 exe1000.py 2015 203
(9, 188)
$ python3 exe1000.py 2015 13
(155, 0)
$
```

https://shell.cloud.google.com/

```python
import sys
x1,x2 = int(sys.argv[1]), int(sys.argv[2])

def divisãoInteira(D,d):
    q = 0       #contador
    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador
    return (q,D)

print(divisãoInteira(x1,x2))

```

```
$ python3 exe1000.py 2015 13
(155, 0)
$
```

https://shell.cloud.google.com/

```python
import sys
x1,x2 = int(sys.argv[1]), int(sys.argv[2])

def div_int(D,d):
    q = 0       #contador
    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador
    return (q,D)

print(div_int(x1,x2))
```

```
$ python3 exe1000.py 2015 17
(118, 9)
$
```

https://shell.cloud.google.com/

Exemplo 2

Exemplo 2

O algoritmo de Euclides

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- O Algoritmo de Euclides (300 a.C.) calcula o Máximo Divisor Comum (MDC) de dois números inteiros positivos, sendo pelo menos um deles diferente de zero.
- O algoritmo usa dois fatos:
  - $MDC(x, 0) = x$
  - $MDC(x, y) = MDC(y, x \,\%\, y)$
- Exemplo:
  - $MDC(21, 15) = MDC(15, 21 \,\%\, 15) = MDC(15, 6)$
  - $MDC(15, 6) = MDC(6, 15 \,\%\, 6) = MDC(6, 3)$
  - $MDC(6, 3) = MDC(3, 6 \,\%\, 3) = MDC(3, 0)$
  - $MDC(3, 0) = 3$

```python
import sys
x1,x2 = int(sys.argv[1]), int(sys.argv[2])

def div_int(D,d):
    q = 0       #contador
    while (D >= d):
        D -= d    # diminuir o Divisor
        q += 1    # aumentar o contador
    return (q,D)
```

```
$
```

https://shell.cloud.google.com/

```
4   def div_int(D,d):
5       q = 0       #contador
6       while (D >= d):
7           D -= d    # diminuir o Divisor
8           q += 1    # aumentar o contador
9       return (q,D)
10
11  def algEuclides(a,b):
12      while(b > 0):
13          (q, r) = div_int(a,b)
14          a = b
15          b = r
16      return a
17
18  print(algEuclides(x1,x2))
```

- O algoritmo usa dois fatos:
  - $MDC(x, 0) = x$
  - $MDC(x, y) = MDC(y, x \,\%\, y)$
- Exemplo:
  - $MDC(21, 15) = MDC(15, 21 \,\%\, 15) = MDC(15, 6)$
  - $MDC(15, 6) = MDC(6, 15 \,\%\, 6) = MDC(6, 3)$
  - $MDC(6, 3) = MDC(3, 6 \,\%\, 3) = MDC(3, 0)$
  - $MDC(3, 0) = 3$

```
$ python3 exe1001.py 15 21
3
$
```

https://shell.cloud.google.com/

```
4   def div_int(D,d):
5       q = 0       #contador
6       while (D >= d):
7           D -= d    # diminuir o Divisor
8           q += 1    # aumentar o contador
9       return (q,D)
10
11  def algEuclides(a,b):
12      while(b > 0):
13          r = div_int(a,b)[1]
14          a = b
15          b = r
16      return a
17
18  print(algEuclides(x1,x2))
```

```
$ python3 exe1001.py 15 21
3
$
```

https://shell.cloud.google.com/

```
4   def div_int(D,d):
5       q = 0       #contador
6       while (D >= d):
7           D -= d    # diminuir o Divisor
8           q += 1    # aumentar o contador
9       return (q,D)
10
11  def algEuclides(a,b):
12      while(b > 0):
13          a,b = b, div_int(a,b)[1]
14
15
16      return a
17
18  print(algEuclides(x1,x2))
```

```
$ python3 exe1001.py 15 21
3
$ python3 exe1001.py 5 25
5
$
```

https://shell.cloud.google.com/

```
4   def div_int(D,d):
5       q = 0       #contador
6       while (D >= d):
7           D -= d    # diminuir o Divisor
8           q += 1    # aumentar o contador
9       return (q,D)
10
11  def algEuclides(a,b):
12      while(b > 0):
13          a,(q,b) = b, div_int(a,b)
14
15
16      return a
17
18  print(algEuclides(x1,x2))
```

```
$ python3 exe1001.py 5 25
5
$ python3 exe1001.py 15 21
3
$
```

https://shell.cloud.google.com/

```
4   def div_int(D,d):
5       q = 0       #contador
6       while (D >= d):
7           D -= d    # diminuir o Divisor
8           q += 1    # aumentar o contador
9       return (q,D)
10
11  def algEuclides(a,b):
12      while(b > 0):
13          a,b = b, div_int(a,b)[1]
14          print("MDC({},{})=".format(a,b),end="")
15
16      return a
17
18  print(algEuclides(x1,x2))
```

```
$
```

https://shell.cloud.google.com/

```
4   def div_int(D,d):
5       q = 0       #contador
6       while (D >= d):
7           D -= d    # diminuir o Divisor
8           q += 1    # aumentar o contador
9       return (q,D)
10
11  def algEuclides(a,b):
12      while(b > 0):
13          a,b = b, div_int(a,b)[1]
14          print("MDC({},{})=".format(a,b),end="")
15
16      return a
17
18  print(algEuclides(x1,x2))
```

```
$ python3 exe1001.py 15 21
MDC(21,15)=MDC(15,6)=MDC(6,3)=MDC(3,0)=3
$ python3 exe1001.py 21 15
MDC(15,6)=MDC(6,3)=MDC(3,0)=3
$
```

https://shell.cloud.google.com/

```
4   def div_int(D,d):
5       q = 0       #contador
6       while (D >= d):
7           D -= d    # diminuir o Divisor
8           q += 1    # aumentar o contador
9       return (q,D)
10
11  def algEuclides(a,b):
12      while(b > 0):
13          print("MDC({},{}%{})=".format(b,a,b),end="")
14          a,b = b, div_int(a,b)[1]
15          print("MDC({},{})=".format(a,b),end="")
16      return a
17
18  print(algEuclides(x1,x2))
```

```
$
```

https://shell.cloud.google.com/

```
4   def div_int(D,d):
5       q = 0       #contador
6       while (D >= d):
7           D -= d    # diminuir o Divisor
8           q += 1    # aumentar o contador
9       return (q,D)
10
11  def algEuclides(a,b):
12      while(b > 0):
13          print("MDC({},{}%{})=".format(b,a,b),end="")
14          a,b = b, div_int(a,b)[1]
15          print("MDC({},{})=".format(a,b),end="")
16      return a
17
18  print(algEuclides(x1,x2))
```

```
$ python3 exe1001.py 21 15
MDC(15,21%15)=MDC(15,6)=MDC(6,15%6)=MDC(6,3)=MDC(3,6%3)=MDC(3,0)=3
$ python3 exe1001.py 210 15
MDC(15,210%15)=MDC(15,0)=15
$
```

https://shell.cloud.google.com/

MCD(10,24)

10

24

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

10

$24 - 10$

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

10

14

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

10

14

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

10

$14 - 10$

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

10

4

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

10

4

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

$10 - 4$

4

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

6 

4 

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

6 

4 

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

$6 - 4$

4

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

2

4

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

2 

4 

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

2 

$4 - 2$ 

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

2

2

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

2 

2 

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

2 

$2 - 2$ 

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

MCD(10,24)

2

0

https://pt.wikipedia.org/wiki/Algoritmo_de_Euclides

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

```
def mdc2(x, y):
  while (y != 0):
    r = x % y
    x = y
    y = r
  return x
```

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

```
1 def mdc2(x, y):
2   while (y != 0):
3     (x, y) = (y, x % y)
4   return x
```

Exemplo 3

Exemplo 3

Mínimo Múltiplo Comúm,
menor múltiplo comum entre dois ou mais números.

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

```
1 def mmc2(x, y):
2
3
4   resultado = 1
5   while (resultado % x != 0) or (resultado % y != 0):
6     resultado = resultado + 1
7   return resultado
```

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

```
1 def mmc2(x, y):
2
3
4   resultado = x
5   while resultado % y != 0:
6     resultado = resultado + x
7   return resultado
```

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

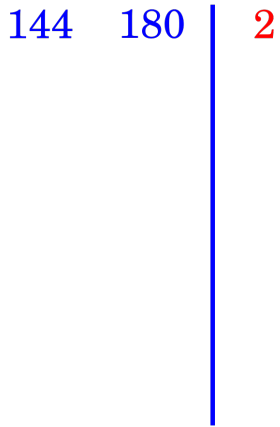
```
1 def mmc2(x, y):
2
3
4   resultado = max(x, y)
5   while resultado % min(x, y) != 0:
6     resultado = resultado + max(x, y)
7   return resultado
```

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

```
1 def mmc2(x, y):
2   if (x < y):
3     (x, y) = (y, x)
4   resultado = x
5   while resultado % y != 0:
6     resultado = resultado + x
7   return resultado
```

144 180


144 180
2

144	180	2
72	90	

144	180	2	2
72	90		

144	180	2	2
72	90		

144	180	2	2
72	90	2	

144	180	2	2
72	90	2	
36	45		

144	180	2	2
72	90	2	2×2
36	45		

144	180	2	2
72	90	2	2 x 2
36	45		

144	180	2	2
72	90	2	2×2
36	45	2	

144	180	2	2
72	90	2	2 x 2
36	45	2	
18	45		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	
9	45		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	
3	15		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15	3	

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15	3	
1	5		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15	3	3 x 3 x 2 x 2 x 2 x 2
1	5		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15	3	3 x 3 x 2 x 2 x 2 x 2
1	5		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15	3	3 x 3 x 2 x 2 x 2 x 2
1	5	5	

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15	3	3 x 3 x 2 x 2 x 2 x 2
1	5	5	
1	1		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15	3	3 x 3 x 2 x 2 x 2 x 2
1	5	5	5 x 3 x 3 x 2 x 2 x 2 x 2
1	1		

144	180	2	2
72	90	2	2 x 2
36	45	2	2 x 2 x 2
18	45	2	2 x 2 x 2 x 2
9	45	3	3 x 2 x 2 x 2 x 2
3	15	3	3 x 3 x 2 x 2 x 2 x 2
1	5	5	5 x 3 x 3 x 2 x 2 x 2 x 2 = 720
1	1		

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

```
def mmc2(x, y):
  resultado = 1
  divisor = 2
  while (x != 1) or (y != 1):
    if (x % divisor == 0) or (y % divisor == 0):
      resultado = resultado * divisor
      if x % divisor == 0:
        x = x / divisor
      if y % divisor == 0:
        y = y / divisor
    else:
      divisor = divisor + 1
  return resultado
```

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

```
1 def mmc2(x, y):
2   return int((x * y) / mdc2(x, y))
```

Exemplo 4

Exemplo 4

Factorial de um número: n!

https://ic.unicamp.br/~mc102/aulas/aula08.pdf

- Possível Resposta:

```
1 def fatorial(x):
2   fat = 1
3   for i in range(1, x + 1):
4     fat = fat * i
5   return fat
6
7 def combinacoes(m, n):
8   return fatorial(m) / (fatorial(m - n) * fatorial(n))
```

Perguntas ....

Referências

- Zanoni Dias, MC102, Algoritmos e Programação de Computadores, IC/UNICAMP, 2021. https://ic.unicamp.br/~mc102/

  Aula Introdutória [slides] [vídeo]
  Primeira Aula de Laboratório [slides] [vídeo]
  Python Básico: Tipos, Variáveis, Operadores, Entrada e Saída [slides] [vídeo]
  Comandos Condicionais [slides] [vídeo]
  Comandos de Repetição [slides] [vídeo]
  Listas e Tuplas [slides] [vídeo]
  Strings [slides] [vídeo]
  Dicionários [slides] [vídeo]
  Funções [slides] [vídeo]
  Objetos Multidimensionais [slides] [vídeo]
  Algoritmos de Ordenação [slides] [vídeo]
  Algoritmos de Busca [slides] [vídeo]
  Recursão [slides] [vídeo]
  Algoritmos de Ordenação Recursivos [slides] [vídeo]
  Arquivos [slides] [vídeo]
  Expressões Regulares [slides] [vídeo]
  Execução de Testes no Google Cloud Shell [slides] [vídeo]
  Numpy [slides] [vídeo]
  Pandas [slides] [vídeo]

- Panda - Cursos de Computação em Python (IME -USP) https://panda.ime.usp.br/

  Como Pensar Como um Cientista da Computação https://panda.ime.usp.br/pensepy/static/pensepy/
  Aulas de Introdução à Computação em Python https://panda.ime.usp.br/aulasPython/static/aulasPython/

- Fabio Kon, Introdução à Ciência da Computação com Python http://bit.ly/FabioKon/
- Socratica, Python Programming Tutorials http://bit.ly/SocraticaPython/
- Google - online editor for cloud-native applications (Python programming) https://shell.cloud.google.com/
- w3schools - Python Tutorial https://www.w3schools.com/python/
- Outros, citados nos Slides.