

# Exercícios (para serem entregues)

Agora é sua vez: faça os exercícios para praticar os conceitos vistos na aula sobre comandos de repetição.

Lembre-se de **salvar** sempre o seu notebook. Ele deverá ser **entregue pelo tidia** para que você receba sua nota!

## Questão 0 (exemplo)

Faça um programa que imprima as  $n$  primeiras potências de 2, em ordem crescente.

**Entrada:** O programa deve receber um único número inteiro  $n$  tal que  $n \geq 0$ .

**Saída:** A resposta consistirá em  $n$  linhas, cada uma com um número que é potência de 2.

```
In [4]: n = int(input("Digite o valor de n: "))
# a primeira potência de 2 é 2 elevado a zero
for i in range(n):
    print(2**i)
```

Digite o valor de n: 3

1  
2  
4

## Questão 1

Faça um programa que mostre os números pares entre 1 e  $n$ , inclusive.

**Entrada:** O programa deve receber um inteiro  $n > 1$ .

**Saída:** Imprima todos os números pares entre 1 e  $n$ , inclusive, se for o caso, um em cada linha.

```
In [37]: n = int(input("Digite um número inteiro: "))

if (n > 1) and (n % 2 == 0):
    for i in range(2, n + 1, 2):
        print(i)

else:
    if (n > 1) and (n % 2 == 1):
        for i in range(2, n, 2):
            print(i)
    else:
        print("Fim")
```

```
Digite um número inteiro: 13
2
4
6
8
10
12
```

## Questão 2

Faça um programa que leia  $n$  valores. Estes valores serão somente negativos ou positivos (nunca nulos). A seguir, mostre a quantidade de valores positivos que foram digitados.

**Entrada:** O programa deve receber inicialmente um inteiro  $n$  ( $n \geq 0$ ). Em seguida, deve receber  $n$  números reais não nulos.

**Saída:** A resposta consistirá de uma única linha, contendo a mensagem `X valores positivos`, onde `X` deve ser corretamente substituído pela quantidade de valores positivos que foram lidos.

```
In [11]: n = int(input("Digite a quantidade de números: "))

positivos = 0

for i in range(n):
    numero = int(input("Entre com o número: "))
    if numero > 0:
        positivos = positivos + 1
print("A quantidade de números positivos informado: ",positivos)
```

```
Digite a quantidade de números: 5
Entre com o número: -2
Entre com o número: 0
Entre com o número: -1
Entre com o número: 3
Entre com o número: 4
A quantidade de números positivos informado:  2
```

## Questão 3

Suponha que seu computador consiga executar somente operações de soma e subtração. Escreva um programa que calcule o produto  $ab$ .

**Entrada:** O programa deverá receber dois números inteiros,  $a$  e  $b$ , respectivamente, com  $a \geq 0$  e  $b \geq 0$ .

**Saída:** A saída consistirá de uma única linha, contendo um único número, resultado de  $ab$ .

In [ ]:

### Questão 4

Faça um programa que calcule a média, o maior número e o menor número de  $n$  números inteiros.

**Entrada:** O programa deve receber inicialmente um inteiro positivo  $n$ , com  $n \geq 1$ , e em seguida deve receber  $n$  números inteiros.

**Saída:** A resposta consistirá de uma única linha, contendo um número real que representa a média dos  $n$  números dados, seguido por dois números inteiros que são o maior e o menor número, respectivamente, dos  $n$  números dados.

In [ ]:

### Questão 5

Faça um programa que leia um inteiro  $n$ , com  $0 \leq n \leq 80$ , e imprima uma saída da forma:

```
1
 2
 3
 4
 .
 .
 .
n
```

*Dica:* pense primeiro em como você imprimiria uma das linhas (uma linha qualquer).

In [ ]:

### Questão 6

Escreva um programa que leia um valor inteiro  $N$ . Este  $N$  é a quantidade de linhas de saída que serão apresentadas na execução do programa.

**Entrada:** O programa deve receber um único inteiro  $N$  com  $N > 0$ .

**Saída:** Imprima a saída conforme o exemplo abaixo.

| Entrada | Saída esperada |
|---------|----------------|
| 7       | 1 2 3 PUM      |
|         | 5 6 7 PUM      |
|         | 9 10 11 PUM    |
|         | 13 14 15 PUM   |
|         | 17 18 19 PUM   |
|         | 21 22 23 PUM   |
|         | 25 26 27 PUM   |

In [ ]:

### Questão 7

Faça um programa que teste se um dado número é primo. Um número  $n$  é primo se os seus únicos divisores são o 1 e o próprio  $n$ .

**Entrada:** O programa deve receber um único número  $n$ , com  $n \geq 2$ .

**Saída:** A resposta consistirá de uma única linha, contendo a palavra `SIM` , se  $n$  for primo, e `NAO` , caso contrário.

*Dica:* Como você resolveria esse problema "na mão"? Pegue uma folha de papel e, sem pensar em programação, observe como você faz para decidir se o número 431, por exemplo, é primo ou não. Pense em todos os números que são candidatos a serem divisores de  $n$  que não sejam o 1 e nem o próprio  $n$ . Se algum deles de fato for divisor de  $n$ , então  $n$  não é primo.

In [ ]:

### Questão 8

Faça um programa que imprima as tabuadas de todos os números entre 1 e  $n$ .

**Entrada:** O programa deve receber o valor de  $n$ .

**Saída** Para cada número  $x$  entre 1 e  $n$ , imprima 11 linhas: a primeira contendo o texto `Tabuada do x:` seguida por 10 linhas, cada uma contendo o texto  $x * i = xi$  , onde  $i$  vai variar entre 1 e 10 e  $x$  ,  $i$  e  $xi$  devem ser devidamente substituídos.

In [ ]:

### Questão 9

Faça um programa que imprima a quantidade de divisores de um número.

**Entrada:** O programa deve receber um único número inteiro  $n$ , com  $n \geq 1$ .

**Saída:** A resposta deverá ser ser no formato `O número de divisores de n é x.` , onde  $n$  e  $x$  devem ser devidamente substituídos.

In [ ]:

### Questão 10

A série de Fibonacci é a seguinte: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, \ldots. Ou seja, o  $i$ -ésimo número da sequência,  $f(i)$ , é dado pela seguinte fórmula:

$$f(i) = \begin{cases} 1 & \text{se } i = 1 \\ 1 & \text{se } i = 2 \\ f(i - 1) + f(i - 2) & \text{se } i > 2 \end{cases}$$

Faça um programa que imprima os  $n$  primeiros números da série de Fibonacci.

**Entrada:** O programa deverá receber um único inteiro  $n$ , com  $n \geq 0$ .

**Saída:** A resposta consistirá em  $n$  linhas, cada uma contendo um número que faz parte da série de Fibonacci.

In [ ]:

