



ALGORITMOS E PROGRAMAÇÃO I



Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

01 |

Implemente um programa em Java que **calcule a soma** dos números naturais de 1 até um número informado pelo usuário.

Entrada	Saída
5	A soma dos números de 1 á 5 é 15

Função Contar

Nomeação

somarAte(numero)

Chamada

somarAte(5) = 15



02 |

Implemente um programa em Java que **calcule o fatorial** de um número informado pelo usuário.

Entrada	Saída
5	O fatorial de 5 é 120

Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

Função Contar

Nomeação

fatorial(numero)

Chamada

fatorial(5) = 120



Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

03 |

Implemente um programa em Java que **calcule a soma dos números naturais pares** de 1 até um número informado pelo usuário.

Entrada	Saída
5	A soma dos números pares de 1 á 5 é 6
10	A soma dos números pares de 1 á 10 é 30

Função Contar

Nomeação

`somarParesAte(numero)`

Chamada

`somarParesAte(5) = 6`



Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

04 |

Implemente um programa em Java que **calcule a soma dos números naturais pares**, a partir de um número de início e fim, informados pelo usuário.

Entrada	Saída
5 10	A soma dos números pares de 5 á 10 é 24

Função Contar

Nomeação

`somarParesAte(inicio, fim)`

Chamada

`somarParesAte(5, 10) = 24`



05 |

Implemente um programa em Java que **calcule a soma dos números naturais múltiplos** de um número informado pelo usuário, a partir de um número de início e fim, também informados pelo usuário.

*Um número (A) é múltiplo de outro (B) quando em sua divisão (A / B) o resto é zero.

Entrada	Saída
4 5 15	A soma dos números múltiplos de 4, de 5 á 15 é 20

Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

Função Contar

Nomeação

```
somarMultiplos(numero, inicio, fim)
```

Chamada

```
somarMultiplos(4, 5, 15) = 20
```



06 |

Implemente um programa em Java que **calcule uma potenciação apenas para números naturais**. O usuário deve informar o número base e o expoente e o programa deve apresentar o resultado da potenciação.

- * Os números naturais são os números inteiros não negativos. $N = \{ 0, 1, 2, 3, \dots \}$
- * Todo número elevado à um expoente 0, tem como resultado 1.
- * Caso o usuário informe qualquer número negativo, deve-se exibir: “Cálculo Inválido”.
- * Você não pode usar a função `Math.pow()`.

Entrada	Saída
2 3	2 elevado a 3 é 8
2 0	2 elevado a 0 é 1
-2 0	Cálculo inválido
2 -1	Cálculo inválido

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Contar

Nomeação

`potenciaN(numero, expoente)`

Chamada

`potenciaN(2, 3) = 8`



07 |

Implemente um programa em Java que **separe os caracteres de um texto** com um hífen (-). O usuário deve informar uma mensagem e o programa deve apresentar a mensagem separando seus caracteres por hífen.

Entrada	Saída
Bora codar	B-o-r-a- -c-o-d-a-r-

Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

Função Contar

Nomeação

separar(texto)

Chamada

separar("Bora codar") =
"B-o-r-a- -c-o-d-a-r-"



08 |

Implemente um programa em Java que **separe os caracteres de um texto** com um hífen (-). O usuário deve informar uma mensagem e o programa deve apresentar a mensagem separando seus caracteres por hífen, seguindo as regras:

- * Espaços devem ser desconsiderados.
- * O último caractere não deve ter hífen.
- * Todos caracteres devem estar em maiúsculo.

Entrada	Saída
Bora codar	B-O-R-A-C-O-D-A-R

Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

Função Contar

Nomeação

```
separar(texto)
```

Chamada

```
separar("Bora codar") =  
"B-O-R-A-C-O-D-A-R"
```



09 |

Implemente um programa em Java que **inverte um texto** informado pelo usuário

Entrada	Saída
Bora codar	radoc aroB
arara	arara

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Contar

Nomeação

`inverter(texto)`

Chamada

`inverter("arara") = "arara"`



10 |

Implemente um programa em Java que **verifique** se um texto informado pelo usuário **é um palíndromo**.

Entrada	Saída
Bora codar	Não é um palíndromo.
arara	É um palíndromo.

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Contar

Nomeação

```
palindromo(texto)
```

Chamada

```
palindromo("arara") = true
```



11 |

Implemente um programa em Java que considerando a sequência de Fibonacci, **identifique** o número de uma posição nessa sequência. O usuário deve informar a posição desejada e o programa deve apresentar o valor encontrado na posição em questão.

* A sequência de Fibonacci inicia-se em $F = \{ 1, 1, \dots \}$ onde o próximo número calculado pela soma dos dois últimos. $F = \{ 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots \}$

Entrada	Saída
10	O valor na posição 10 da sequência de Fibonacci é 55
5	O valor na posição 5 da sequência de Fibonacci é 5
1	O valor na posição 1 da sequência de Fibonacci é 1
2	O valor na posição 2 da sequência de Fibonacci é 1
3	O valor na posição 3 da sequência de Fibonacci é 2

Implemente o exercício ao lado, seguindo a estrutura de funções abaixo

Função Contar

Nomeação
`fibonacci(posicao)`

Chamada
`fibonacci(10) = 55`



12 |

Implemente um programa em Java que **verifique** se um número é primo.

O início da sequência dos números primos são:
P = { 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, ... }

- * Um número primo é aquele que só é divisível por 1 por ele mesmo.
- * Um número (A) é divisível por outro (B), quando o resto de (A / B) é igual a zero.
- * Os números 0 e 1 não são primos.

Entrada	Saída
11	É um número primo.
37	É um número primo.
12	Não é um número primo.
1	Não é um número primo.

Implemente o exercício ao lado,
seguindo a estrutura de funções abaixo

Função Contar

Nomeação

ehPrimo(numero)

Chamada

ehPrimo(11) = true



Bons estudos!
Bruno de Oliveira