

Instituto Federal de Educação Ciência e Tecnologia do RN - IFRN

Exercícios

001 Escreva uma função que retorne o maior de 2 números. O nome da função deve ser **obrigatoriamente** `maior2`, como especificado abaixo:

```
def maior2 (a,b)
  # Corpo da funcao
end
```

002 Escreva uma função que retorne o maior de 5 números. O nome da função deve ser **obrigatoriamente** `maior5`, como especificado abaixo:

```
def maior5 (a,b,c,d,e)
  # Corpo da funcao
end
```

003 Escreva uma função, chamada `quantidade_de_vogais`, que receba uma string e retorne a quantidade de vogais da string. A função deve verificar se o parâmetro é do tipo string e, caso, não seja, a função deve retornar `-1`.

```
def quantidade_de_vogais (s)
  # Corpo da funcao
end
```

004 Escreva uma função, chamada `impar?` (a interrogação faz parte do nome da função), que receba um valor e retorne `true` se o mesmo é ímpar, caso contrário a função retorna `false`.

```
def impar? (n)
  # Corpo da funcao
end
```

005 Escreva uma função que retorne a quantidade de palavras de uma *string*. A função **DEVE** verificar se o parâmetro passado é uma *string*.

```
def conta_palavras (texto)
  # Corpo da funcao
end
```

006 Escreva uma função que receba dois número inteiros *a* e *b* e retorne um *array* com todos os números entre *a* e *b*, inclusive.

```
def intervalo_a_b (n1,n2)
  # Corpo da funcao
end
```

007 Escreva uma função que receba uma matriz (*array* de *array*) e retorne a soma de todos os elementos da matriz.

```
def soma_elementos_matriz (m)
  # Corpo da funcao
end
```

008 Escreva uma função que receba um número e retorne o fatorial do mesmo.

```
def fatorial (n)
  # Corpo da funcao
end
```

009 Escreva uma função que receba um número inteiro e retorne um *array* com os divisores do número, incluindo o 1 e ele mesmo.

```
def divisores (n)
  # Corpo da funcao
end
```

010 Escreva uma função que receba um número inteiro e verifique se o mesmo é um número primo, retornando **true** ou **false**.

```
def primo? (n)
  # Corpo da funcao
end
```

011 Escreva uma função que receba dois números inteiros e retorne **true** se os mesmo são **primos entre si** ou **false** caso contrário.

```
def primos_entre_si? (n1, n2)
  # Corpo da funcao
end
```

012 Escreva uma função que calcule o MDC (Máximo Divisor Comum) de dois números dados.

```
def mdc (n1, n2)
  # Corpo da funcao
end
```

013 Escreva uma função que, dado dois números, retorne o MMC dos dois.

```
def mmc (n1, n2)
  # Corpo da funcao
end
```

014 Escreva uma função, chamada **transcreve_numero(num)**, que transcreva os dígitos de um dado número de telefone. A função deve receber uma *string* contendo um número telefonico e deve retornar uma lista (*array*) dos nomes dos dígitos. Qualquer caractere que não seja número deve ser ignorado. Considerando que o conteúdo da *string* passada seja "3221-1234" a função deve retornar o *array* ["tres", "dois", "dois", "um", "um", "dois", "tres", "quatro"].

```
def transcreve_numero (telefone)
  # Corpo da funcao
end
```

015 Escreva uma função que receba duas matrizes e identifique as coincidências entre elas, ou seja, os elementos que se encontram na mesma posição (linha e coluna). A função deve retornar uma nova matriz contendo os elementos que coincidem nas suas posições e 0 (zero) nas demais posições. Todas as matrizes recebidas pela função e a matriz retornada devem ter o mesmo número de linhas e colunas.

```
def coincidencia_matriz (m1, m2)
  # Corpo da funcao
end
```

016 Escreva uma função receba como parâmetro um *array* de números inteiros e retorne um *array* com a maior **sub-lista** ordenada crescentemente. Uma sub-lista ordenada crescentemente é uma lista de números inteiros ordenados que fazem parte da lista recebida como parâmetro. Por exemplo: se a função recebe como parâmetro a lista [8,9,1,7,8,17,3] então a maior sub-lista é a [1,7,8,17], pois esta lista possui 4 elementos. As sub-listas ordenadas crescentemente neste *array* são: [8,9], [1,7,8,17] e [3].

```
def sublista_ordenada (a)
  # Corpo da funcao
end
```

017 Escreva uma função que receba uma matriz e verifique se a mesma é **triangular superior**.

```
def triangular_superior? (matriz)
  # Corpo da funcao
end
```

018 Escreva uma função que calcule a massa de uma molécula a partir de sua especificação.

As moléculas são identificadas através de uma sequência de caracteres, representando o símbolo do elemento, e, possivelmente o número logo após o símbolo. Por exemplo, a molécula da água é representada através da sequência H2O, onde H é o símbolo do Hidrogênio, o 2 informa a quantidade de átomos desse elemento e o O é o símbolo do Oxigênio. Como não há número após o oxigênio existe apenas um átomo de oxigênio. Através da massa atômica do hidrogênio e do oxigênio é possível calcular a massa da molécula da água, da seguinte forma:

$$\begin{aligned} \text{massa}_{H_2O} &= \text{massa}_H \times 2 + \text{massa}_O \\ \text{massa}_{H_2O} &= 1 \times 2 + 16 \\ \text{massa}_{H_2O} &= 18 \end{aligned}$$

OBS: A massa do oxigênio é de 15,9994 mas para simplificação de cálculo considere um número inteiro com a massa aproximada.

A função deve verificar se o tipo passado é uma *string* que contenha uma molécula válida. A função deve retornar apenas um inteiro que seja a massa da molécula passada. Se o parâmetro passado não for uma molécula válida a função deve retornar -1 . A tabela abaixo mostra exemplo de parâmetros e retornos da função.

Parâmetro	Retorno
H2O	18
CO2	44
C12H22O11	342

```
def massa_molecula (molecula)
    # Corpo da funcao
end
```

019 Escreva uma função que receba uma matriz quadrada e verifique se a mesma forma um quadrado mágico (http://pt.wikipedia.org/wiki/Quadrado_mágico). A sua função deve verificar:

1. Se o parâmetro passado é uma matriz
2. Se a matriz é quadrada
3. Se TODOS os elementos são números inteiros

A função deve retornar **true** ou **false**.

```
def quadrado_magico? (matriz)
    # Corpo da funcao
end
```

020 Determinado jogo representa um labirinto através de uma matriz de caracteres com NL linhas e NC colunas. Em cada posição i, j da matriz está armazenado um caractere "P" para indicar uma parede, " " (espaço) para representar caminho aberto e a posição em que o jogador está é determinada por um "*" (asterisco). Pedese que você crie a função `move_direita(labirinto)` que altera o labirinto de forma que o jogador se movimenta à direita, caso possa. Obviamente, o jogador só pode se mover à direita, caso haja um espaço em branco a direita do local onde ele se encontra. A função deve retornar **true** se o jogador foi movimentado e **false**, caso contrário. O labirinto é uma matriz.

```
def move_direita(labirinto)
    # Corpo da funcao
end
```

Exemplo: Caso a função receba o labirinto abaixo passado como parâmetro ela deve retornar **false**

```
[
  ["P", " ", " ", " "],
  ["P", "*", "P", " "],
  ["P", "P", "P", " "],
]
```

Exemplo: Caso a função receba o labirinto abaixo passado como parâmetro ela deve retornar **true**

```
[  
  ["P", "*", " ", " "],  
  ["P", " ", "P", " "],  
  ["P", "P", "P", " "],  
]
```

e modificar a matriz para:

```
[  
  ["P", " ", "*", " "],  
  ["P", " ", "P", " "],  
  ["P", "P", "P", " "],  
]
```