

Exercícios
Aula 10 – Funções

1. Faça um programa que leia a ordem (tamanho de linhas e colunas) de uma matriz quadrada. Em seguida, leia números para os elementos desta matriz. Por fim, especifique a função *imprime_matriz(matriz, ordem)*, que deverá imprimir todos os elementos da matriz recebida por parâmetro.

2. Faça um programa para ler e somar duas matrizes 5 x 4 (matriz A e matriz B). A soma deve ser realizada na função *soma_matriz(matA, matB)*, sendo que a mesma deverá retornar uma nova matriz contendo o resultado da soma. Por fim, deve-se imprimir a matriz retornada pela função no código principal do programa.

3. Faça um programa igual ao exercício 2. Porém, a leitura das matrizes deve ser realizada utilizando uma função que deve ser chamada duas vezes, uma para a matriz A e outra para a matriz B. Além disso, a impressão da soma das matrizes A e B também deve ser realizada em uma função.

4. Implemente um programa que possua uma função que recebe como parâmetro um número inteiro, e a partir dele imprime a sua tabuada. Se o número recebido for 2, esta função deve imprimir:

Tabuada do 2:

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$2 \times 4 = 8$$

$$2 \times 5 = 10$$

$$2 \times 6 = 12$$

$$2 \times 7 = 14$$

$$2 \times 8 = 16$$

$$2 \times 9 = 18$$

$$2 \times 10 = 20$$

Na código principal do programa, faça a chamada desta função para exibir as tabuadas do 2 ao 12.

5. Faça um programa para calcular o resultado da expressão $(x^2 + 3xy + y) / (2xy + 3x + 4y + 2)$ para os valores $x = 2, 4, 6, 8$ e $y = 1, 3, 5, 7$. O resultado da expressão deve ser calculado em uma função, sendo retornado por ela. Como saída, devem ser impressos os valores calculados para todos os valores de x e y .

6. Faça um programa que leia dois números, A e B, que representam um intervalo numérico. Após isso, para cada número do intervalo, chame uma função (que deve ser implementada) que retorna True caso o número seja primo, e False caso contrário. A saída do programa deve imprimir todos os números primos que existem no intervalo.

7. Implemente um programa que possui uma função chamada *divisao_inteira(n1, n2)*, sendo que a mesma deve dividir $n1$ por $n2$ utilizando apenas operadores de soma e subtração, retornando o resultado desta operação (somente parte inteira, desconsiderando o resto). Ao executar o programa,

ele deve ler dois números (N1 e N2) e calcular sua divisão inteira através da chamada da função previamente implementada. Obs.: a função deve tratar divisão por 0.

8. Sem utilizar a operação de multiplicação, escreva um programa que leia dois números inteiros e multiplique-os. Por exemplo: $2 * 4 = 2 + 2 + 2 + 2$. O cálculo da multiplicação deve ser implementado dentro de uma função.

9. Faça uma função que receba dois números inteiros como parâmetros. A função retorna um inteiro que é a quantidade de números primos existentes entre os dois números informados como parâmetro. Obs. pode-se utilizar a função definida no exercício 6 para identificar se um número é primo.

10. Faça uma função que receba um vetor de inteiros de tamanho arbitrário. A função inverte os elementos do vetor, colocando-os de trás para frente. Ex.: se o vetor de entrada foi 2, 10, 9, 5, 3 a função irá alterar o vetor para 3, 5, 9, 10, 2. **Obs.: não utilizar o método reverse().**

11. Faça uma função que receba dois vetores de inteiros, chamados A e B, sendo A e B com tamanho de 10 elementos. A função cria um novo vetor C, e coloca nele todos os valores que existem nos vetores A e B, porém em ordem crescente. Ex.: (assumindo que A e B possuem tamanho 3): A {20, 3, 11} e B {5, 13, 2}, a função produzirá o vetor C {2, 3, 5, 11, 13, 20}. O vetor C deverá ser retornado pela função e impresso no código principal do programa. **Obs.: pode utilizar o método sort().**

12. Faça uma função que receba dois vetores de inteiros, chamados A e B, sendo A e B com tamanho de 10 elementos. A função cria um novo vetor C, e coloca nele todos os valores que existem no vetor A, porém não existem no vetor B. No fim o vetor C deve ter 10 elementos. Caso isso não aconteça, preencha com 0 as posições até completar o vetor com os 10 elementos. **Obs.: não utilize o método count().**