

## Funções

1) Faça um programa em C que contenha uma **função** denominada SOMATORIO que realize o seguinte calculo:

$$S = 1/1 + 2/3 + 3/5 + 4/7 + 5/9 + \dots + n/m.$$

A função deve receber como parâmetro de entrada um valor inteiro (n) e deve retornar o valor resultante da operação.

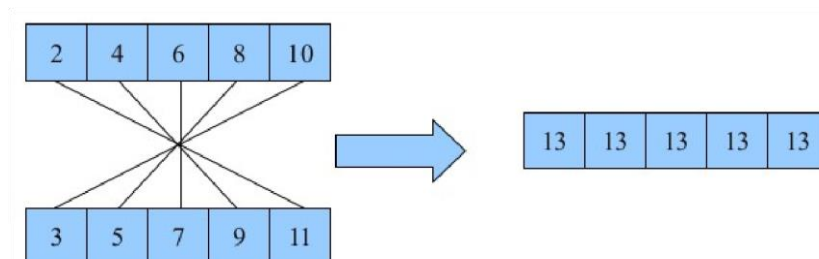
2) Faça um programa que contenha uma **função** chamada NUMEROS\_PRIMOS, essa função deve apresentar ao usuário uma lista dos números primos existentes entre 1 e um número inteiro (NUM) informado pelo usuário.

A função não deve ter retorno e deve receber um valor inteiro (NUM) como parâmetro de entrada da função.

## Vetores

3) Faça um programa em C que realize a soma de dois **vetores** de inteiros (A e B). A soma deve ser feita de tal forma que o primeiro elemento do vetor A soma com o último elemento do vetor B, armazenando o resultado na primeira posição do vetor C, como no exemplo abaixo. Os valores resultantes da soma dos dois vetores devem ser armazenados em um terceiro vetor de inteiros (C).

Ex.:



Obs.: Os vetores devem ser alocados dinamicamente.

Os vetores devem apresentar 10 posições.

O programa deve funcionar corretamente para quaisquer valores (Não somente para o exemplo acima).

Os resultados das operações devem ser armazenados nas posições sequenciais (0, 1, 2, ...).

4) Crie um programa que realize a busca de um dado elemento inteiro em um **vetor** de inteiros. Os valores inseridos no vetor devem ser solicitados do usuário. O programa deve pesquisar pelo elemento e imprimir o valor, a posição e a quantidade de ocorrências deste valor no vetor. Caso o valor não seja encontrado, uma mensagem deve informar ao usuário que tal valor não existe no vetor.

Obs.: O vetor deve ser alocado dinamicamente.

O tamanho do vetor deve ser solicitado do usuário.

O programa deve funcionar corretamente para quaisquer valores.

### Matrizes

5) Desenvolva um programa em C que realize a soma de duas **matrizes** inteiras de tamanho (5x5) (A e B). A soma de duas matrizes inteiras consiste em realizar a adição dos elementos correspondentes das matrizes, ou seja, soma do elemento da primeira linha e primeira coluna de A com o elemento da mesma linha e coluna de B, este resultado é armazenado na mesma posição da matriz C resultante, como no exemplo abaixo.

2	4	6
8	10	12
14	16	18

1	3	5
7	9	11
13	15	17

3	7	11
15	19	23
27	31	35

A correspondência dentre os pontos pode ser melhor compreendida através das cores das matrizes.

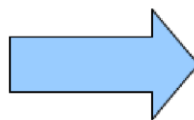
Obs.: As matrizes devem ser alocadas dinamicamente.

6) Crie um programa em C que realize a troca dos elementos da diagonal principal pelos elementos da diagonal secundária. Para isso deve ser criada uma **matriz** de inteiros (A) (5x5) que deve ser preenchida pelo usuário. Os elementos: 2, 10 e 18, correspondem aos elementos da diagonal principal, enquanto os elementos: 6, 10 e 14, correspondem aos elementos da diagonal secundária. A diagonal secundária pode ser melhor compreendida na figura abaixo.

2	4	6
8	10	12
14	16	18

A troca dos elementos está ilustrada na figura abaixo.

2	4	6
8	10	12
14	16	18



6	4	2
8	10	12
18	16	14

Na figura acima o elemento 2 da diagonal principal trocou de posição com o elemento 6 da diagonal secundária. O elemento 10 não trocou de posição pois pertence tanto a diagonal principal quanto a diagonal secundária e o elemento 14 da diagonal principal trocou de posição com o elemento 18 da diagonal secundária.

Obs.: A matriz deve ser alocada dinamicamente.

## Estruturas

7) Faça um programa em C para cadastrar produtos, as informações a serem inseridas são:

- Nome do produto;
- Quantidade no estoque;
- Preço;
- Fornecedor.

Obs.: O programa deve utilizar **struct**.

O programa deve ser capaz de cadastrar 50 produtos.

O programa deve armazenar as informações de cada produto e mostrar tais informações.

8) Faça um programa em C que resolva o seguinte problema. Você foi contratado por uma empresa para desenvolver um sistema e deve criar um programa que cadastre a empresa e seus funcionários (15 funcionários), as informações a serem cadastradas são:

Endereço

- Rua;
- Número.

Funcionário

- Nome do funcionário;
- Cargo que exerce;
- Informações de endereço (Rua e Número).

Empresa

- Nome da empresa;
- CNPJ;
- Informações de funcionário (Nome, Cargo, Endereço do funcionário).

Obs.: O programa deve utilizar **struct**.

O programa deve criar 3 estruturas (Endereco, Funcionario, Empresa).

O programa deve armazenar as informações e mostrar tais informações.

## Recursividade

9) Faça um programa em C que contenha uma função que realize a pesquisa binária em um vetor (**recursivamente**).

Obs.: Pesquisar como funciona a “pesquisa binária”.

10) Faça um programa em C que contenha uma função para:

- Apresentar na tela os  $n$  primeiros elementos da sequência de Fibonacci;
- Somar dos 10 primeiros termos da série de Fibonacci. Apresentar o resultado na tela.

Obs.: Pesquisar como funciona a “série de Fibonacci”.  
O programa deve utilizar **recursividade**.