

Lista 5**Funções; Recursão;****Questão 1**

Escreva uma função que recebe dois pontos $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ e retorna a distância euclidiana entre esses pontos, ou seja:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
1 float distancia(int x1, int y1, int x2, int y2);
```

Em seguida, escreva um programa que receba como entrada os dois pontos e imprima a distância utilizando a função criada.

Questão 2

Escreva uma função que testa se um número i é primo.

```
1 int ehPrimo(int i);
```

Depois, escreva um programa que receba como entrada um valor inteiro n e em seguida imprima todos os números primos de 1 até n utilizando a função `ehPrimo()` criada.

Questão 3

Escreva uma função que recebe dois números inteiros positivos a e b por parâmetro e determina se eles são amigos ou não, devolvendo 1 caso sejam amigos, e 0 caso contrário.

```
1 int amigos(int a, int b);
```

Dois números são amigos se cada número é igual à soma dos divisores próprios do outro (os divisores próprios de um número m são os divisores estritamente menores que m).

Exemplo:

Os divisores próprios de 220 são 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110, cuja soma é 284; e os divisores próprios de 284 são 1, 2, 4, 71 e 142, cuja soma é 220. Logo, 220 e 284 são números amigos.

Questão 4

Escreva um função que receba um vetor de inteiros **A** e o seu tamanho **n** por parâmetro e retorna o vetor invertido.

```
1 int inverta(int A[], int n);
```

Tente resolver esse problema sem utilizar nenhum vetor auxiliar.

Questão 5

Escreva uma função que receba dois vetores *A* e *B* e o seus respectivos tamanhos por parâmetro verifica se os valores de *B* estão contidos (em qualquer ordem em *A*).

```
1 int compara(int A[], int tamA, int B[], int tamB);
```

Questão 6

Escreva uma função que receba um vetor de números reais (**float**) **F** e o seu tamanho **n** por parâmetro e devolva a média aritmética dos números do vetor.

```
1 float media(float F[], int n);
```

Questão 7

O que faz a função abaixo?

```
1 void imprime_alguma_coisa(int n){
2     if (n != 0) {
3         imprime_alguma_coisa(n / 2);
4         printf("%c", '0' + n % 2);
5     }
6 }
```

Escreva um programa para testar a função `imprime_alguma_coisa()`.

Questão 8

Escreva uma função **recursiva** que computa o fatorial de um número *n* passado por parâmetro.

```
1 long long int fat(int n);
```

Questão 9

Quem faz mais operações: a versão iterativa ou a versão recursiva do fatorial? Os dois usam a mesma quantidade de memória?

Questão 10

Escreva uma função **recursiva** que calcula a potência a^b para valores *a* e *b* passados por parâmetro (não use bibliotecas como `math.h`).

```
1 double pot(int a, int b);
```