

### Lista de Exercícios – Ordenação

**Questão 1:** Faça um programa que leia n nomes inserindo-os em uma lista de forma ordenada utilizando a ideia do algoritmo **Insertion Sort**. No final, o programa deve mostrar todos os nomes ordenados alfabeticamente.

**Questão 2:** Crie um programa que dado uma string, coloque as letras dela em ordem decrescente usando o algoritmo **Quick Sort**.

**Questão 3:** Considere a seguinte estrutura:

```
struct pessoa{  
    int Matricula;  
    char Nome[30];  
    float Nota;  
};
```

Faça uma função que dado um array de tamanho N dessa estrutura, ordene o array pelo campo escolhido pelo usuário. A ordenação deve utilizar o método **Merge Sort**.

**Questão 4:** Este desafio consiste em calcular 100 alturas de alunos geradas aleatoriamente entre 1-3 metros com até duas casas decimais. Essas alturas servirão como parâmetro para a encomendada de bebedouros de alturas específicas para conseguir atender todas as pessoas eficientemente, da menor até a maior. Para a programação, você deve utilizar a ordenação **BubbleSort**.

Nesse desafio deve-se mostrar todas as alturas coletadas e como saída, deve-se mostrar todas as alturas ordenadas em ordem crescente. Além disso, para o cálculo dos números aleatórios deve-se usar as bibliotecas e 2 comandos:

```
#include<time.h>
```

```
#include<stdlib.h>
```

```
srand(time(NULL));
```

Essa função gera um número aleatório de acordo com o tempo.

```
n+rand()%k;
```

Essa função diz o alcance dos números sorteados: de n ate k

**Questão 5:** Faça um programa que leia n nomes e ordene-os pelo tamanho utilizando o algoritmo **Selection Sort**.