



### Atividade Prática

**Assunto :** Ordenação – Bolha e Bolha Inteligente

**Questão 1:** Um dos algoritmos mais simples de ordenação é o chamado método BOLHA, cujo algoritmo é apresentado na Figura 1.

---

**Algoritmo: ORDENAÇÃO - BOLHA**

---

**Entrada:** Vet: Vetor de números naturais; Tam: Tamanho do vetor (inteiro)

**Saída:** Vet ordenado em ordem crescente

```
1 início
2   para i ← 0 até Tam faça
3     para j ← 1 até Tam faça
4       se Vet[j] < Vet[j - 1] então
5         aux ← Vet[j - 1]
6         Vet[j - 1] ← Vet[j]
7         Vet[j] ← aux
8       fim
9     fim
10  fim
11 fim
```

---

**Figura 1 :** Algoritmo de ordenação pelo método bolha.

Uma variação significativa do método bolha é o chamado **Bolha Inteligente**, que realiza menos comparações que o Bolha.

Nessa atividade vocês deverão implementar os métodos Bolha e Bolha Inteligente em C, seguindo as orientações abaixo:

- O projeto deve ser modularizado (main.c, bolha.c, bolha.h). Considere os arquivos main.c e bolha.h disponibilizados. Não é permitido fazer alterações neles. Todas as funções presentes no bolha.h devem ser implementadas.
- Esse exercício é também para exercitarem o tratamento de dados em C e revisitarem os conceitos de struct e TAD. Para tanto, considere o tipo de dado composto definido pelo struct tReturn da Figura 2. Ele deve ser obrigatoriamente implementado no arquivo bolha.c.
- No moodle:
  - Sua resposta deve ser **APENAS** o upload do arquivo bolha.c (esse deve ser o nome do arquivo).



```
struct tReturn  
{  
    int *vet;  
    int tam;  
    int erro;  
};
```

**Figura 2:** Tipo tReturn.

Para essa atividade, o vetor é lido a partir de um arquivo. Na Figura 3 há um exemplo de arquivo. Nesse exemplo, o arquivo chama 'entradaAleatoria.txt'. A primeira linha contém a quantidade de elementos do vetor (5). As demais, os elementos que devem compor o vetor. Portanto, na Figura 3, em a) há o exemplo de como é o arquivo de entrada; em b) de como deve ficar o vetor preenchido.

entradaAleatoria.txt						
1	5					
2	3					
3	1					
4	2					
5	4					
6	7					

a)

3	1	2	4	7
---	---	---	---	---

b)

**Figura 3:** Exemplo do arquivo de entrada de dados.

Na função *main*, o nome do arquivo de entrada é lido e então a função *leArquivo* é chamada. Essa função constrói uma instância do tipo tReturn descrito na Figura 2. No caso do arquivo apresentado na Figura 3, o retorno da função *leArquivo* seria como ilustrado abaixo.

tReturn varRetorno:

vet : 

3	1	2	4	7
---	---	---	---	---

tam: 5

erro: 0

Após compor a variável do tipo tReturn, a função *main* chama os algoritmos de ordenação Bolha e depois o Bolha Inteligente para realizar a ordenação do vetor.