

**AULA PRÁTICA 07 – ALGORITMOS DE ORDENAÇÃO - HEAPSORT**

- Data de entrega: Até 16 de Abril às 23:55.

- Procedimento para a entrega:.

1. Submissão: via **Moodle**.
2. Os nomes dos arquivos e das funções devem ser especificados considerando boas práticas de programação.
3. Funções auxiliares, complementares aquelas definidas, podem ser especificadas e implementadas, se necessário.
4. A solução deve ser devidamente modularizada e separar a especificação da implementação em arquivos *.h* e *.c* sempre que cabível.
5. Os arquivos a serem entregues, incluindo aquele que contém *main()*, devem ser compactados (*.zip*), sendo o arquivo resultante submetido via **Moodle**.
6. Dentre os arquivos submetidos, deve existir um intitulado *compilcao.txt*, contendo os comandos especificados no *prompt/console* para compilar e executar seu programa.
7. Caracteres como acento, cedilha e afins não devem ser utilizados para especificar nomes de arquivos ou comentários no código.

- Bom trabalho!

## Questão

Implemente o algoritmo de ordenação *HeapSort*. Seu algoritmo receberá um vetor de inteiros e o tamanho deste vetor e retornará o vetor ordenado, o número de comparações de chaves e movimentos de registros realizados durante sua execução

**Entrada e saída:** A entrada inicia com o número de vetores a serem ordenados. Cada vetor é iniciado pelo número de elementos contidos no vetor e é seguido dos elementos a serem ordenados.

A saída apresenta o vetor ordenado, a quantidade de comparações e movimentos realizados.

### Exemplo de entrada e saída

Entrada	Saída
2	0 1 2 3 4 5 6 7 8 9 10
11 0 1 2 3 4 5 6 7 8 9 10	0 1 2 3 4 5 6 7 8 9 10
11 10 9 8 7 6 5 4 3 2 1 0	Vetor   Comparacoes   Movimentacoes
	1.....  C1   M1
	2.....  C2   M2

Onde C1 é o número de comparações feitas na execução do algoritmo para o vetor 1 e M1 o número de movimentações. O mesmo vale para C2 e M2 só que para o vetor 2.

A entrega deve ser feita até 16 de Abril às 23:55 pelo Moodle em um arquivo *.zip*.