Escrever um programa para fazer ordenação de um vetor de floats. Esse programa deve ser capaz de ordenar usando os métodos insertion, shell e quicksort. Deve também ser capaz de fazer buscas por determinados elementos no vetor.

A entrada do programa é um conjunto de valores lidos de um arquivo (vetor.dat) e um segundo conjunto de valores lidos de um outro arquivo (busca.dat). A escolha pelo algoritmo de ordenação e pelo método de busca deve ser feito a partir de um menu.

A saída do programa deve ser um arquivo (resultado.dat) contendo uma linha para cada valor buscado, sendo que o conteúdo dessa linha, quando o valor for encontrado, deve ser da forma (não escrever os colchetes na saída):
[índice da posição encontrada] [valor contido naquela posição] [valor buscado]

Se não for encontrado o valor, então a linha deverá ser da forma (não escrever os colchetes na saída):
[-1] [0.0] [valor buscado]

Como os valores são reais, então se considera como valor encontrado se:
| valor - vetor[i] | < 0.0001

Você deverá incluir no seu programa a medição do tempo gasto nas tarefas de ordenação, quando for o caso, e de busca, considerando que as seguintes medidas deverão ser fornecidas:

1. Tempo de ordenação
2. Tempo de busca
3. Tempo total (ordenação mais busca)

Os casos a serem avaliados são:

1. Busca sem ordenação
2. Ordenação por insertion e busca sequencial
3. Ordenação por shell sort e busca sequencial
4. Ordenação por quicksort e busca sequencial
5. Ordenação por insertion e busca binária
6. Ordenação por shell sort e busca binária
7. Ordenação por quicksort e busca binária

Para a medição dos tempos use a função clock_gettime(CLOCK_REALTIME, struct timespec *)

Entregue o código de seu(s) programa(s) e um relatório com a análise sobre os tempos medidos (média dos tempos de 5 execuções para cada caso). Observe que são 3 conjuntos diferentes de arquivos, que devem ser medidos separadamente para gerar curvas de tempo.

Limites de valores:
vetor.dat pode ter até 500 mil elementos
busca.dat pode ter até 100 mil elementos
-10000000.0 < vetor[i] < 10000000.0, com 6 casas decimais

Os arquivos para teste foram postados em separado.