

Teste 2 – Busca em vetor de Estruturas

Considere que os dados dos atletas de um clube são armazenados em um vetor de estruturas, contendo nome, altura e peso de cada atleta, ordenado em ordem crescente de nome e altura (o peso não importa para a ordenação). Para não haver problemas de precisão, considere alturas e pesos como números inteiros.

Escreva uma função que use a técnica de **busca binária** para apenas encontrar e retornar a posição (sem realizar nenhuma inserção) onde um novo atleta deve ser inserido, mantendo a ordenação. Por exemplo, a estrutura {ada, 170, 60} deve ser inserida após a posição 2, e {ada, 160, 60} antes do início. Use obrigatoriamente uma **função auxiliar de comparação**.

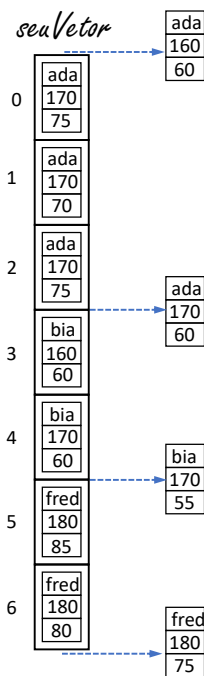


Fig.1

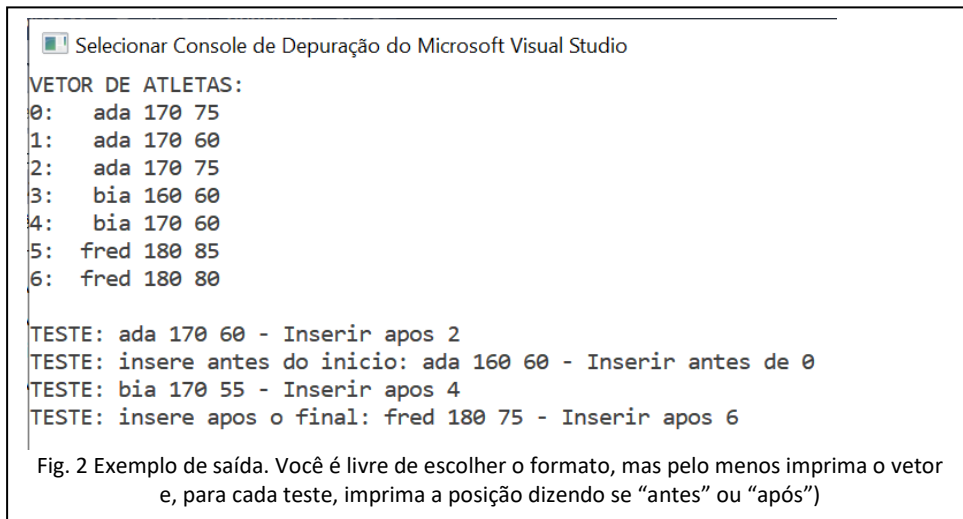


Fig. 2 Exemplo de saída. Você é livre de escolher o formato, mas pelo menos imprima o vetor e, para cada teste, imprima a posição dizendo se “antes” ou “após”)

Considere que o vetor pode ter mais de um atleta com o mesmo par de nome e altura e que, neste caso, a posição de um novo atleta com os mesmos dados deve ser após o último atleta igual. Veja exemplos na Fig. 1 e Fig. 2, onde {ada, 170, ...} se repete 3 vezes e {fred, 180, ...} se repete 2 vezes. O vetor pode ter qualquer tamanho (no exemplo da Fig.1, o vetor pode ser dimensionado para muito mais do que 7 elementos).

Escreva obrigatoriamente uma função auxiliar que monte o vetor. Você pode montar o vetor da maneira que você achar mais fácil (lendo ou não de arquivo e alocando memória estaticamente ou dinamicamente, tanto faz). Mas, se optar por ler de arquivo, entregue o arquivo de entrada como .txt (caso contrário o teste é anulado). Não é permitido ler do teclado. Nomeie arquivos com o seguinte formato:

nome_ultimoSobrenome_matricula_Teste02_G2.

Teste sua função com situações similares às exibidas na Fig. 2. Use obrigatoriamente valores completamente diferentes dos usados nos exemplos acima, mas não menos de 7 elementos no vetor (e não use o mesmo número de repetições). Inclua testes de inserção antes do início e depois do final.

Dica: Uma vez encontrado um atleta igual, percorra sequencialmente o vetor para pular os iguais. Observe o que retorna da função de comparação e observe o valor das variáveis do algoritmo para você encontrar a posição correta da inserção (sugestão: retorne -1 se for para inserir antes do início).

DISCLAIMER:

Por favor coloque as seguintes linhas de comentário no início do seu código (contendo os seus dados), como sendo a sua identificação e a sua declaração:

```
/*
NOME COMPLETO:
MATRICULA PUC-Rio:
DATA:
DISCIPLINA: INF1007 TURMA (33A, 33B, 33C, 33D):
DECLARACAO DE AUTORIA:
    Declaro que este documento foi produzido em sua totalidade por mim,
    sem consultas a outros alunos, professores ou qualquer outra pessoa.
*/
```