

Teste 3 – Arquivos e Vetores de Ponteiros

Considere que um determinado sistema de coleta de dados por sensores gera dois arquivos textos, onde cada linha destes arquivos diz respeito a um conjunto de sensores. O primeiro arquivo, denominado **rotulos.txt**, contém os nomes dos sensores e o segundo arquivo, denominado **valores.txt**, contém os valores registrados por estes sensores (como números inteiros). Para facilitar o processamento, cada linha do arquivo **rotulos.txt** se inicia com o número de sensores do conjunto em questão. Além disso, cada linha dos arquivos está ordenado crescentemente pelo valor registrado. Considere que o tamanho dos nomes de sensor são todos menores do que um determinado tamanho arbitrado por você. No exemplo a seguir, há dois conjuntos de sensores, um com 7 sensores e outro com 12 sensores:

rotulos.txt

```
7 a10 a20 a30 a40 a50 a60 a70
12 b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12
```

valores.txt

```
2 4 4 5 7 11 12
3 4 4 4 7 10 11 12 14 16 17 18
```

Escreva e teste um programa que lê estes dois arquivos textos e grava um novo arquivo binário, chamado **eliminados.dat**, que contém os rótulos dos sensores cujos valores são menores do que a **mediana** do seus conjuntos. No exemplo acima (veja arquivo **valores.txt**), como a mediana do primeiro conjunto é 5.0 é a mediana do segundo conjunto é 10.5, o arquivo binário corresponde aos seguintes rótulos:

a10 a20 a30 b1 b2 b3 b4 b5 b6.

onde os valores 2, 4 e 4 correspondem aos a10, a20 e a30 e os valores 3, 4, 4, 4 e 7 correspondem aos b1, b2, b3, b4, b5 e b6. Como o arquivo **eliminados.dat** é binário, as nove *strings* estão gravadas (em bytes) com um caractere após o outro (até chegar ao EOF), sem brancos:

```
a 1 0 \0 a 2 0 \0 a 3 0 \0 ... b 4 \0 b 5 \0 b 6 \0 EOF
```

Considere obrigatoriamente as seguintes restrições:

- Ao ler o primeiro arquivo, coloque os rótulos de cada conjunto em um mesmo vetor de ponteiros para strings, alocado dinamicamente e do tamanho exato. Libere este vetor tão logo seja possível.
- Ao ler o segundo arquivo, coloque os valores em um mesmo vetor de inteiros, alocado dinamicamente e do tamanho exato. Libere este vetor tão logo seja possível.
- Teste com rótulos, valores e número de conjuntos completamente diferentes do exemplo acima.
- Cuide de todas as situações que requerem o aborto da execução, liberações de memória e fechamentos de arquivo.
- Escreva obrigatoriamente uma função auxiliar que calcula e retorna a **mediana** de um vetor de inteiros.
- No teste, imprima o que está lendo. Imprima as medianas também.
- Teste a gravação do arquivo binário lendo-o e imprimindo cada uma das *strings* contidas nele.

Deposite um arquivo **.c** e os dois arquivos .txt usados. Nomeie os arquivos com o seguinte formato:

nome_ultimoSobrenome_matricula_Testes03

Dica: o cálculo da mediana requer que o conjunto esteja ordenado. Se há um número ímpar de elementos, a mediana é o elemento do meio e, se há um número par de elementos, a mediana é a média dos dois números no meio do conjunto.

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):
DECLARACAO DE AUTORIA:
Declaro que este documento foi produzido em sua totalidade por mim,
sem consultas a outros alunos, professores ou qualquer outra pessoa.
*****/
```