

**Primeiro Trabalho Prático**

**Introdução**

O atual prefeito de Alvinópolis city, o sr. João Louco, está querendo criar inovações no processo eleitoral e para isso necessita de um sistema para realização das eleições, onde serão computados os votos das pessoas e apuração dos resultados.

Tendo em vista os possíveis milhões de votantes e não permitir que um eleitor vote com o mesmo título mais de uma vez, você deverá utilizar uma estrutura de dados que possibilite checar se o voto de um determinado título já foi computado ou não, num tempo muito curto. Adicionalmente, caso o voto não tenha ainda sido computado, a estrutura deverá possibilitar a inserção desse voto no sistema em um tempo muito curto. Por fim, o sistema deve permitir excluir votos suspeitos (pessoas que votam no lugar de votantes que faltam, a estrutura deverá possibilitar remoção de um voto corrupto em tempo novamente baixíssimo).

Mas que estruturas fantásticas são essas, que permitem inserção, remoção, contagem de elementos e checagem de elementos numa velocidade bastante alta? Como bom aluno de EDII, você identifica imediatamente que estamos nos referindo a tabelas hash e árvores de pesquisa.

**Algoritmos e Estruturas**

Primeiramente, implemente o TAD cédula de eleição, com campos pra votos para prefeito e vereador(inteiros) e título de eleitor(string).

Em seguida, implemente as seguintes estruturas que serão utilizadas no processo de contabilização dos votos:

1. Tabela hash (com resolução de colisão por hashing duplo).
2. Tabela hash (com resolução de colisão por árvore binária de pesquisa).
3. Tabela hash (com resolução de colisão por lista encadeada).

Todas estruturas devem permitir inserção, remoção e checagem da existência de elementos.

Por fim, implemente duas tabelas hash (com resolução de colisão por endereçamento aberto) para servir de apuração dos votos a qualquer momento da eleição (uma para prefeito e outra para vereador).

**Restrições**

1. O código deve ser escrito em C.
2. A documentação do TP deve conter no máximo 8 páginas.
3. O seu código deve compilar usando C padrão ANSI. Basta evitar utilizar funções específicas do Windows ou Linux.
4. O seu código deve compilar sem nenhuma warning! Você pode verificar se seu código compila sem warnings com o seguinte comando no Linux  
`gcc -Werror <arquivos.c>`

**Entrada**

Seu programa irá ler da entrada padrão (usando scanf normal).

A primeira linha contém o identificador de estrutura: “1” para tabela hash (com resolução de colisão por hashing duplo); “2” para tabela hash (com resolução de colisão por árvore binária de pesquisa) e “3” para tabela hash (com resolução de colisão por lista encadeada).

A terceira linha contém o número de candidatos a vereador, prefeito e número de eleitores. As linhas seguintes irão conter um dos códigos das ações com seus possíveis argumentos, de acordo com a tabela a seguir:

Ação	Código	Parâmetros	Exemplo
------	--------	------------	---------

Votar	0	Classe do candidato, título de eleitor, número do candidato	0 1 d42cc 50
Remover	1	Título do eleitor	1 i1f3e3t7
Apurar	2	Classe do candidato, tamanho do ranking	2 0 4
Encerrar	3	-	3

A tabela abaixo mostra os possíveis valores para os parâmetros

Parâmetro	Valores possíveis
Classe do candidato	0: prefeito 1: vereador
Título do eleitor	Até 8 caracteres, com letras minúsculas e/ou números
Número do candidato	Até 8 caracteres, apenas números
Tamanho do ranking	Inteiro de 1 até 100

Assim, o programa deve ficar lendo continuamente e processando as ações, até que leia a entrada “3”, correspondendo ao comando encerra.

Dicas:

1. Ao computar um voto, deve-se checar se o título já votou antes para o cargo em questão.
2. Ao remover um título, deve-se remover os votos aos quais ele estava vinculado.
3. Não necessariamente todos os eleitores vão votar, nem todos candidatos receberão pelo menos um voto.
4. Use scanf e printf na fase de implementação. Na fase de teste use fscanf e fprintf com os arquivos de testes que serão passados.

### Saída

Os resultados da saída devem ser armazenados em arquivo texto(na fase de implementação use a saída padrão(usando o printf normalmente). Cada comando de entrada deve exibir uma saída diferente, conforme as regras a seguir:

Ação votar: imprimir “<voto computado, candidato X tem Y voto(s) | voto não computado>”

Exemplo:

```
0 0 abbf3 40
voto computado, candidato 40 tem 132 voto(s)
```

```
0 0 abbf3 30
voto não computados
```

Aqui, o voto não foi computado, desde que o eleitor “abbf3” já tinha votado para presidente.

Ação remover: imprimir mediante <removido|nãoremovido>, X voto(s) válidos no sistema.”

Exemplo:

```
1 133calote
mediante removido, 456 voto(s) válidos no sistema.
1 3honesto
mediante não removido, 567 voto(s) válido(s) no sistema.
```

Aqui, o cidadão não foi removido, pois ele ainda não tinha votado.

Ação apurar: imprimir “<numero do cnadidao , X voto(s)>” para os Y primeiros candidatos(de

acordo com o parâmetro da ação), ordene-os pelo número de votos, um por linha. Em caso de empate, ordenar pelo número do candidato em ordem crescente, Exemplo:

2 0 4

13 3241 voto(s).

25 2301 voto(s)

40 1056 voto(s).

45 0 voto(s).

Encerra: apenas fecha o programa

## **Documentação**

Escreva um documento explicando seu código(e não apenas colocando trechos dele) e avaliando o desempenho de sua implementação. Separe em 5 seções.

### 1 – Introdução

Escreva uma pequena introdução, dando uma ideia geral do sistema e de cada um dos algoritmos e estruturas usadas.

### 2 – Implementação

Descreva os algoritmos implementados.

### 3 – Experimentos

Você deverá rodar cada um dos 3 algoritmos para as bases de testes que serão repassadas para vocês. Apresente o tempo de execução em gráficos ou tabelas. Também compute o número de colisões de cada algoritmo para os arquivos de teste.

Responda:

- a) Qual dos algoritmos é mais rápido para cada base de teste?
- b) Quais foram as funções de hash que você usou? Por que as escolheu?

### 4- Conclusões

Resuma o que você fez e deixou de fazer neste TP. Explique quais foram as dificuldades que você encontrou para fazer este TP.

### 5 – Referências

Cite as fontes utilizadas para realização deste TP, mesmo que sejam sites da internet.

Obs.:

- a) O trabalho pode ser feito em dupla;
- b) Data de Entrega: 01/09/2020;
- c) Caprichem no código fonte: comentários na dose certa, identificação e modularização;
- d) Comecem logo pois a data de entrega jamais estará tão distante como agora.