

Prova de IP-EC
Prof Alexandre Mota - 19/09/2022

Questão. Como estamos em época de eleição, que tal criarmos um sistema para computar votos? O sistema deve ler os nomes e números dos candidatos a partir de um arquivo texto (separados por vírgula, como exemplificado em sala de aula). Do mesmo modo com os eleitores: nome e CPF. O sistema deve perguntar pelo CPF do eleitor, verificar que pertence ao arquivo e só então permitir a votação (Mensagem de erro deve ser exibida, caso contrário). A votação seguirá esta ordem: deputada(o) federal, deputada(o) estadual, senador(a), governador(a) e presidente. Para encerrar o processo de votação, o CPF -1 deve ser informado ao sistema. Neste momento, o sistema deverá apresentar os candidatos que receberam votos (apenas estes) e o quantitativo de votos. Para modularizar, use as seguintes funções:

- `listaCandidatos carregarCandidatos(char *nomeArq)`: esta função deve ler os dados (nome e número) dos candidatos do arquivo de nome `nomeArq` e devolver estrutura `listaCandidatos`;
 - estrutura `listaCandidatos` deve conter 5 ponteiros para a estrutura `Candidato`, um ponteiro para deputada(o) federal, deputada(o) estadual, senador(a), governador(a) e presidente. Além disso, deve conter um vetor (`qtdCandidatos`) de tamanho 5 para guardar a quantidade de candidatos de cada um dos 5 cargos (Por exemplo: `qtdCandidatos[0]` refere-se à dep federal, `qtdCandidatos[1]` à dep estadual e assim por diante);
 - estrutura `Candidato` deve conter um nome e um inteiro com seu número;
- `Eleitor *carregarEleitores(char *nomeArq, int *qtdEleitores)`: esta função deve ler os dados (nome e CPF) dos eleitores válidos, a partir do arquivo de nome `nomeArq` e devolver vetor dinâmico da estrutura `Eleitor`, alterando o ponteiro `qtdEleitores` para figurar o tamanho do vetor dinâmico;
- `int verificaCandidato(listaCandidatos listaC, Candidato candidatoAt)`: esta função deve verificar se `candidatoAt` pertence à lista `listaC`, retornando 1 em caso afirmativo e 0 em caso negativo;
- `int verificaEleitor(Eleitor *listaE, Eleitor eleitorAt)`: esta função verifica se `eleitorAt` pertence à lista `listaE`, retornando 1 em caso afirmativo e 0 em caso negativo;
- `void apresentaResultado(TotalVoto *votos, int qtdCandidatos)`: esta função deve simplesmente percorrer o vetor dinâmico `votos` (até o total de `qtdCandidatos`) e mostrar na tela este conteúdo, um a um.
 - estrutura `TotalVoto` deve conter um ponteiro pra `Candidato` (que deverá ser preenchido a partir do endereço armazenado na estrutura `listaCandidatos`) e um inteiro com o total de votos que este recebeu;
- `Candidato *localizaCandidato(listaCandidatos listaC, Candidato candidatoAt, char *cargo)`: esta função deve procurar pelo `candidatoAt` no vetor dinâmico correspondente ao cargo, cujos vetores dos vários cargos estão na estrutura `listaC`.

Seu programa principal deve apresentar um menu que basicamente permite votar, apresentar votação atual ou então sair do sistema. Votar entra em laço que só sai com CPF igual a -1. As demais funções acima serão acionadas nos locais esperados (as de carregamento, antes de apresentar o menu).

Boa sorte!!!