

1. Escreva uma função em C que leia um arquivo de entrada no qual, em cada linha, tem-se um nome seguido de um CPF. O nome é separado do CPF por um caractere de tabulação '\t'. A função deverá ler cada nome e armazenar em um vetor de strings, ler o CPF correspondente e armazenar em outro vetor de strings na mesma posição correspondente ao nome. O cabeçalho da função deve ser o seguinte:

```
int le_candidatos(char nomes[][TAM_MAX_NOME+1], char CPFs[][TAM_CPF+1])
```

onde TAM_MAX_NOME é uma constante que define o número máximo de caracteres em um nome qualquer, TAM_CPF é o número de caracteres numéricos em um CPF (igual a 11). A função, internamente, deve solicitar a digitação do nome do arquivo a ser lido. A função retorna o número de candidatos efetivamente lidos. Após efetuar a leitura, a função deve se encarregar de fechar o arquivo de entrada.

2. Escreva uma função em C que leia um arquivo de entrada no qual cada linha é composta por um CPF seguido de NUM_COLS números inteiros não negativos. A função, internamente, deve solicitar a digitação do nome do arquivo a ser lido. A função deverá montar uma matriz com NUM_COLS+1 colunas, na qual a primeira coluna (coluna 0) é para armazenar a soma dos valores lidos e cada uma das demais colunas é reservada para cada um dos números inteiros lidos. A linha utilizada para armazenar os valores é selecionada pelo CPF, isto é: a função deverá procurar pelo CPF em um vetor de strings (fornecido como parâmetro) e armazenar os dois inteiros na linha de mesmo índice desse vetor. O Cabeçalho da função deve ser o seguinte:

```
int le_resultados (int result[][NUM_COLS+1], char CPFs[][TAM_CPF+1])
```

A função retorna o número de linhas lidas e preenchidas na matriz. Caso haja qualquer problema na leitura de qualquer das linhas, a função deverá retornar o valor zero (isso inclui encontrar um CPF inexistente ou um número negativo nas colunas de inteiros).

3. Escreva uma função em C que receba como parâmetro uma matriz de inteiros com NUM_COLS + 1 colunas e coloque suas linhas em ordem decrescente considerando o seguinte critério de ordenação:
 - a. a soma das NUM_COLS últimas colunas (armazenada na coluna 0);
 - b. caso o critério a) resulte em empate de algumas linhas, deve-se usar, como critério de desempate, a coluna 1;
 - c. em caso de empates no critério b) deve-se usar a coluna 2,
 - d. caso ainda persistam empates, esses devem ser resolvidos pela coluna seguinte e assim sucessivamente.

Note que é necessário reordenar também apropriadamente os vetores de nomes e CPFs. O cabeçalho da função deve ser o seguinte:

```
void ordena (int matriz[][NUM_COLS+1],  
            char nomes[][TAM_MAX_NOME+1],  
            char CPFs[][TAM_CPF+1], int tamanho  
            )
```

4. Use as funções anteriores para escrever um programa que leia os nomes e CPFs dos candidatos de um concurso público de um arquivo cujo nome será solicitado.

O concurso tem 30 questões de cunho específico (da área do concurso) e 10 de português.

Após efetuar a leitura dos candidatos, o programa efetua a leitura de um arquivo com a tabela de acertos das questões por candidato. Nesse arquivo cada linha contém três colunas, a primeira com o CPF, a segunda com o número de acertos das questões específicas e a terceira com o número de acertos das questões de português.

O programa deve então criar um arquivo de saída, cujo nome deverá ser solicitado ao usuário, que deverá conter, em ordem, do primeiro para o último colocado, um por linha:

- o nome,
- o CPF,
- o número de acertos total,
- o número de acertos em específicas e
- o número de acertos em português.

A ordenação segue o critério de primeiro levar em conta o número total de acertos, depois o número de acertos em específicas e depois o número de acertos em português.