1. Escreva uma função em C que leia um arquivo de entrada com uma lista de nomes e armazene esses nomes em um vetor de strings. O cabeçalho da função deve ser o seguinte:

```
int le_nomes(char nomes[][TAM_MAX_NOME+1])
```

   onde TAM_MAX_NOME é uma constante que define o número máximo de caracteres em um nome qualquer. A função deve solicitar o nome do arquivo a ser lido. A função retorna o número de nomes 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 de um arquivo uma tabela de números inteiros com um número qualquer de linhas e um número de colunas especificado pela constante NUM_COLS. A função deve solicitar o nome do arquivo a ser lido e armazenar os números em uma matriz. O Cabeçalho da função deve ser o seguinte:

```
int le_tabela(int tabela[][NUM_COLS], int quant)
```

   A primeira coluna da tabela (coluna 0) só deve conter valores não negativos menores do que *quant* (parâmetro da função). Caso contrário, se, durante a leitura do arquivo a função ler um valor maior ou igual a *quant* ou negativo na primeira coluna de qualquer linha, a função deve parar a leitura, fechar o arquivo e retornar o código 0.

   O número de linhas da tabela encontrada no arquivo deve ser exatamente igual a *quant*; caso o número de linhas lidos ultrapasse o valor *quant* ou não chegue a *quant*, a função deve parar a leitura, fechar o arquivo e retornar o código -1.

   A primeira coluna de cada linha também não deve ter um valor igual ao de qualquer outra linha; caso contrário, a função deve parar e retornar um código -2.

   Caso o numero de linhas da tabela seja exatamente igual a quant, e todas as linhas tenham sua primeira coluna entre 0 e *quant*-1 e sejam diferentes entre si, então a função deve retornar o valor 1.

   Após efetuar a leitura, a função deve se encarregar de fechar o arquivo de entrada.

   Teste com NUM_COLS = 6.

3. Escreva uma função em C que receba como parâmetro uma matriz de inteiros com NUM_COL colunas e coloque suas linhas em ordem decrescente considerando a primeira coluna; em caso de empates na primeira coluna, deve-se desempatar utilizando a segunda coluna; caso ainda persistam empates, esses devem ser resolvidos pela terceira coluna e assim sucessivamente. O cabeçalho da função deve ser o seguinte:

```
void ordena (int matriz [][NUM_COLS], int quant)
```

   onde *quant* é o número de linhas utilizado da matriz. Teste com NUM_COLS = 6.

4. Use as funções anteriores para escrever um programa que leia os nomes dos times de um campeonato de futebol de um arquivo cujo nome será solicitado. Um exemplo de conteúdo desse arquivo poderá ser:

```
Flamengo
Fluminense
Santos
Vasco
Corintias
```

Após efetuar a leitura dos nomes dos times, o programa efetua a leitura da tabela da situação do campeonato, na qual encontra-se, em cada linha, uma sequencia de 6 valores inteiros que representam respectivamente:

- o código de um time
- os pontos já ganhos por esse time
- o numero de jogos já disputados por esse time
- o número de vitórias já conquistadas por esse time
- o saldo de gols já obtido por esse time
- e os gols pró já marcados por esse time

Um exemplo para o conteúdo desse arquivo poderá ser:

```
0 10 8 3 -4 12
1 17 8 5 10 19
2 10 8 3 -5 11
3 11 8 3 -1 15
4 19 8 6 13 23
```

A primeira coluna (de código) diz qual o time está sendo especificado, indexando uma das linhas do arquivo de times.

O programa deve então criar um arquivo de saída que deverá conter, em ordem, do primeiro para o último colocado, o nome do time, sua colocação atual, seu número de pontos, o número de jogos disputados, o número de vitórias conquistadas, o saldo de gols e os gols pró. O critério de ordenação segue a prioridade das colunas 1 a 5 da tabela de situação do campeonato, ou seja: primeiro usa-se o número de pontos; em caso de empate, utiliza-se o número de jogos, e assim por diante. A saída para o exemplo acima deverá ser:

```
            Nome| Col | PG  | Jog | Vit | SG  | GP  |
----------------|-----|-----|-----|-----|-----|-----|
       Corintias|   1 |  19 |   8 |   6 |  13 |  23 |
      Fluminense|   2 |  17 |   8 |   5 |  10 |  19 |
           Vasco|   3 |  11 |   8 |   3 |  -1 |  15 |
        Flamengo|   4 |  10 |   8 |   3 |  -4 |  12 |
          Santos|   5 |  10 |   8 |   3 |  -5 |  11 |
```