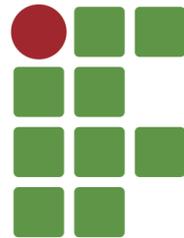


Fórmula 1

Programação de Computadores I

Ciência da Computação

Prof. Daniel Saad Nogueira Nunes



**INSTITUTO
FEDERAL**
Brasília

1 Contextualização

O campeonato mundial de Fórmula 1 é dividido em várias etapas, cada uma disputada em um circuito através do mundo. Temos etapas na Itália (Monza), Mônaco, Brasil (Interlagos), México (Hermanos Rodríguez), Bélgica (Spa-Francorchamps), e em diversos outros países. Os pilotos são pontuados de acordo com a ordem de chegada em cada etapa. Desde 2010, adota-se a Tabela 1 para pontuar os pilotos em cada etapa.

Tabela 1: Pontuação por ordem de chegada na Fórmula-1.

Posição	Pontos
1°	25
2°	18
3°	15
4°	12
5°	10
6°	8
7°	6
8°	4
9°	2
10°	1
>10°	0

Além do campeonato mundial de pilotos, também há o campeonato mundial de construtores, disputado pelas equipes, ou *scuderias*. Nesse caso, a pontuação da equipe corresponde à soma da pontuação de seus dois pilotos.

Ao final do campeonato os pilotos são classificados em ordem decrescente de pontuação, isto é, os pilotos com mais pontos ficam em uma posição mais elevada. Em caso de empate no número de pontos, os pilotos são desempatados pelo número de vitórias: aquele com mais vitórias (maior número de primeiros lugares) tem precedência sobre aquele com menos. Caso o número de pontos e vitórias entre dois pilotos seja igual, utiliza-se o número de segundos lugares. Se o empate persistir, utiliza-se o número de terceiros lugares, e assim em diante. O mesmo critério é utilizado para o campeonato de construtores, mas em vez de um piloto, estão envolvidos os dois da mesma equipe.

Implemente um sistema que, dados os resultados de cada etapa do campeonato mundial de fórmula 1, imprima a classificação final dos pilotos e dos construtores.

2 Especificação

Os caminhos dos arquivos de entrada e saída deverão ser obtidos através da linha de comando.

- Primeiro argumento: caminho do arquivo de entrada.
- Segundo argumento: caminho do arquivo de saída;

2.1 Arquivo de entrada

O arquivo de entrada possui a informação sobre os pilotos de Fórmula 1, as equipes, e o resultado atingido pelos pilotos em cada etapa.

A primeira linha do arquivo de entrada possui dois inteiros N e M , separados por um espaço, que indicam, respectivamente, o número de pilotos e o número de etapas. As próximas N linhas possuem o nome de um piloto e o nome da sua equipe. É garantido que cada equipe tem dois pilotos.

Em seguida, há a descrição dos resultados de cada uma das M etapas. Cada etapa possui N linhas com o nome dos pilotos. A primeira linha da descrição de uma etapa possui o nome do piloto que venceu aquela etapa, a segunda linha, o nome do piloto que chegou em segundo lugar, e assim por diante.

Existe um separador --- entre a descrição dos pilotos e equipes e o início da descrição das etapas e também ao final de cada descrição de uma etapa.

Restrições:

- $2 \leq N \leq 40$;
- $1 \leq M \leq 1000$;
- Cada equipe é composta de dois pilotos.
- O nome dos pilotos e das equipes consistem apenas de letras maiúsculas ou minúsculas e hífen, além de estarem limitados a 30 caracteres. Não há espaços.

2.2 Arquivo de Saída

O arquivo de saída deverá listar os pilotos em ordem de classificação do campeonato, com a posição final, número de pontos, vitórias e pódios (números de vezes em que o piloto chegou na terceira posição ou superior). Cada informação de um piloto está em uma linha, e os dados estão separados por um espaço.

Após a classificação dos pilotos, deverá constar a classificação dos construtores, também em ordem decrescente. Na classificação dos construtores, deverá ser impresso, em cada linha do arquivo de saída, a posição da equipe, seu nome, e sua quantidade de pontos.

Deverá existir um separador --- entre as classificações dos pilotos e equipes.

2.3 Documentação

O código deve ser bem documentado, com presença de comentários explicando os trechos mais complexos do código. Além disso, um arquivo README deve ser providenciado com a devida identificação do autor descrevendo o projeto e instruindo como o código deve ser compilado.

2.4 Modularização

O sistema deverá ser dividido em módulos, cada qual com uma tarefa. Estes módulos podem ser organizados internamente através de várias funções e eles correspondem aos seguintes:

- Módulo de leitura: efetua a leitura do arquivo de pilotos e etapas.
- Módulo de saída: produz a classificação final no arquivo de saída.
- Módulo de processamento: efetua o processamento da pontuação dos pilotos.
- Módulo de ordenação: ordena os pilotos de acordo os critérios da competição.
- Módulo principal: contém a função main e realiza a chamada aos outros módulos.

Os módulos devem ser organizados em arquivos separados, com seus respectivos arquivos de cabeçalho e implementação.

2.5 Construção do sistema

Um `Makefile` deverá ser produzido para a compilação dos códigos-fontes no executável e deverá ser distribuído junto ao código.

2.6 Alocação Dinâmica

Para este projeto, a alocação dinâmica deverá ser utilizada para todas as estruturas de dados, obrigatoriamente. Logicamente, as estruturas alocadas devem ser liberadas no momento propício.

2.7 Exemplos

Arquivo de entrada

```
4 2
Verstappen RBR
Perez RBR
Leclerc Ferrari
Sainz Ferrari
---
Verstappen
Perez
Sainz
Leclerc
---
Perez
Sainz
Leclerc
Verstappen
---
```

Arquivo de saída

```
1 Perez 43 1 2
2 Verstappen 37 1 1
3 Sainz 33 0 2
4 Leclerc 27 0 1
---
1 RBR 80
2 Ferrari 60
---
```

Arquivo de entrada

```
4 3
Verstappen RBR
Perez RBR
Leclerc Ferrari
Sainz Ferrari
---
Verstappen
Sainz
Leclerc
Perez
---
Perez
Sainz
Leclerc
Verstappen
---
Perez
Leclerc
Verstappen
Sainz
---
```

Arquivo de saída

```
1 Perez 62 2 2
2 Verstappen 52 1 2
3 Sainz 48 0 2
4 Leclerc 48 0 3
---
1 RBR 114
2 Ferrari 96
---
```

3 Critérios de correção

Deve ser utilizada a linguagem de programação C para a implementação do projeto.

Para validação da correção do algoritmo, testes automatizados serão realizados, então é **crucial** que a saída esteja conforme o especificado.

Serão descontados pontos dos códigos que não possuírem indentação.

3.1 Ambiente de Correção

Para a correção dos projetos, será utilizada uma máquina de 64-bits com sistema operacional GNU/LINUX e compilador GCC 10.2.0, logo é imprescindível que o sistema seja capaz de ser compilado e executado nesta configuração.

4 Considerações

- GDB, Valgrind e ferramentas gráficas associadas podem ajudar na depuração do código.
- Este trabalho deve ser feito **individualmente**.
- Não serão avaliados trabalhos que não compilem.
- Como parte da correção é automatizada, deverá ser impresso apenas o que a especificação pede. Atentem-se para a formatação da saída.
- A incidência de plágio será avaliada automaticamente com nota 0 para os envolvidos. Medidas disciplinares também serão tomadas.
- O trabalho deve ser entregue dentro de uma pasta zipada com a devida identificação do aluno no prazo combinado pelo ambiente virtual de aprendizagem da disciplina.