(TAD\_opac\_15) Problema: Nesta atividade, vamos simular um sistema baseado nos jogos da franquia Super Trunfo e na coleção de [cards da elma-chips Dracomania](http://colecoes-elma-chips.blogspot.com/2013/01/cards-dracomania.html). A ideia é simples, seu programa deve conter uma lista de cartas com seus atributos, e depois deve simular “lutas” entre essas cartas.

**Cartas**:

Nossas cartas serão baseadas no jogo dracomania (segue exemplo):



Todas as cartas deverão possuir as seguintes informações:

* Identificador
* Nome
* Poder de Magia
* Poder de Força
* Poder de Fogo

Usando como exemplo a primeira carta da imagem acima, temos:

* Nome: Dragao Branco
* Identificador: 7
* Magia: 21
* Força: 15
* Fogo: 24

**Luta:**

Uma luta funcionará comparando duas cartas em um de seus atributos que será selecionado. Em caso de empate a carta com menor identificador deve vencer.

Toda luta deve conter os seguintes atributos:

* Identificador da luta
* Identificadores dos lutadores 1 e 2
* Atributo a ser comparado (M para magia, F para força, G para fogo)

**Regras de funcionamento**

* Seu programa deve ter a capacidade de cadastrar uma carta (C), realizar uma luta (L) ou finalizar o programa (F).
* As cartas cadastradas possuem algumas restrições:
  + Todas terão nome de tamanho máximo de 50 caracteres
  + Nenhum de seus atributos pode possuir valor maior que 30
  + A soma de seus atributos não pode ser maior que 80

**Caso algum dos tópicos não seja válido a carta deve ser desconsiderada.**

* Em caso de cadastro de carta com um ID já existente a carta deve ser atualizada no sistema, lembrando que a atualização requer os mesmos padrões de cadastro definidos anteriormente, ou seja caso tente cadastrar uma carta com um mesmo ID, mas fora dos padrões os valores antigos devem se manter.
* As lutas sempre se darão entre duas cartas válidas, deve se comparar o poder do atributo selecionado para a luta.
* Caso uma resulte em empate, a carta vencedora deve ser aquela com menor valor de ID

Ao finalizar o programa deve-se gerar um relatório final com:

* Número de cartas cadastradas
* Número de cartas invalidadas no cadastro
* Quantidade de lutas
* Quantidade de lutas que foram para critério de desempate (**se houver luta**)
* Nome da carta com maior número de vitórias (**se houver luta**)
* ID da Luta com maior diferença de poder (**se houver luta**)

**Padrão de entrada**

Todas as entradas devem ser lidas a partir da entrada padrão. Cada uma das opções devem ser lidas da seguinte forma:

Cadastro de cartas:

C (caractere que representa o cadastro)

7 (identificador da carta)

Dragao Branco (nome da carta)

21 15 24 (atributos, na sequência magia, força e fogo)

Exemplo de cadastro de Luta:

L (caractere que representa uma luta)

1 (identificador da luta)

7 8 M (ID da carta 1, ID da carta 2 e o atributo de comparação, que pode assumir M, F ou G)

A execução do programa deve acabar ao ler o caractere F.

**Padrão de saída**

A saída deve ser impressa na tela de acordo com o seguinte padrão:

Quantidade de Cartas: 10

Quantidade de descartadas: 2

Quantidade de lutas: 5

Lutas que foram desempatadas: 0

Carta mais vitoriosa: Dragao Vermelho

Luta com Maior Diferença: 5

**Informações adicionais**

O código deve seguir as interfaces definidas nos arquivos ".h" fornecidos junto com o exercício. Esses arquivos não devem ser modificados, pois contêm as especificações do problema a ser resolvido.

Além disso, é crucial enfatizar que você será responsável pela manipulação correta da memória durante a execução do programa. Isso inclui a alocação dinâmica e a liberação de memória conforme necessário, garantindo que não haja vazamentos de memória. Qualquer erro relacionado à manipulação de memória, detectado pelo Valgrind, resultará em um decréscimo na nota atribuída à atividade.

