Trabalho Prático - Treinamento de Pokemon
Disciplina de Algoritmos e Estruturas de Dados I
Curso de Engenharia de Computação
Professores: Bruno Ribas e Silvio Boss

1 O Mundo Pokemon

O mundo dos treinadores `Pokemon` está bem agitado com a aproximação do torneio internacional de batalhas `Pokemon`. Ash está com diversas dificuldades em continuar seu treinamento pois Mist e Brock não ajudam mais como antigamente. Mist resolveu voltar a treinar com suas irmãs e Brock só pensa em seu namoro com a enfermeira Joy de Hyrule. Apesar das dificuldades Ash não desiste e quer se tornar o maior medalhista de todos os tempos com seus `Pokemons` e quer fazer um treinamento rigoroso.

Uma batalha `Pokemon` consiste em uma disputa entre dois `Pokemons` comandados por seus respectivos mestres e por isso o treinador deve conhecer seus Pokemons, e acima de tudo, o treinador deve ser o melhor amigo do `Pokemon` e o `Pokemon` o melhor amigo de seu treinador. Quando essa relação simbiótica está feita as batalhas poderão fluir de uma forma jamais vista.

Para continuar o treinamento você foi contratado para desenvolver um programa de computador que ajude Ash a escolher o `Pokemon` certo e executar o melhor golpe para nocautear o inimigo.

Questão de Implementação

A sua tarefa consiste em desenvolver um programa que dado um conjunto de `Pokemons` com suas respectivas habilidades e Inimigos com suas fraquezas, consiga utilizar o menor número de ataques possível.

---

2 Dados de Entrada

A entrada consiste por diversas linhas que são respectivamente:

- A primeira linha possui um número N ($1 \leq V \leq 100$).
- As N próximas linhas possuem diversos campos, que representam:
  - Nome do `Pokemon` (máximo de 20 caracteres);
  - Uma lista de habilidades no formato [Habilidade1: força, Habilidade2: força,...]
  - Essa lista termina quando o caractere lido for ']'
- Depois de lido todos os `Pokemons` será passada uma lista de inimigos no formato:
  - Nome do `Pokemon` (máximo de 20 caracteres);
  - Lista de fragilidades no formato [Fragilidade1:multiplicador, Fragilidade2: Multiplicador, ...]
  - Essa lista termina quando o caractere lido for ']'
  - Quando nome do Inimigo for **"FIM"** significa que a entrada termina.

Cada Fragilidade de um `Pokemon` inimigo possui o mesmo nome da Habilidade de seu `Pokemon`. O ataque é contabilizado multiplicando a força da habilidade do seu `Pokemon` com o multiplicador da Fragilidade para essa Habilidade. Um inimigo pode ser atacado por apenas 1 `Pokemon` seu e cada inimigo possui 100 de energia.

3 Exemplo

```
2
Pikachu [ energia: 3, grito: 5 ]
Bulbasaur [ grama: 5, chicote: 10 ]

Inimigo1 [ chicote: 3, grito: 2 ]
FIM
```

4 Exemplo de Saída

```
Bulbasaur chicote 4
```

Explicação do Exemplo

No exemplo dado você possui 2 `Pokemons` *(Pikachu e Bulbasaur), e apenas 1 inimigo. Assim, a saída foi escolher o* `pokemon Bulbasaur` *com o poder* ***chicote 4*** *vezes que retira (10 * 3) * 4 = 120 pontos de energia. A escolha* `Pikachu grito 10` *significa que Pikachu utilizou 10 vezes grito para tirar 100 de energia, esse ataque é pior que a saída de exemplo pois Bulbasaur Utiliza apenas 4 golpes de chicote. Quando houver empate, ou seja, duas ou mais habilidades requerem a mesma quantidade de golpes, utilize a que possui a menor ordem lexicográfica.*

---

5 Execução

As avaliações feitas pelos professores será feita em Linux e por isso os códigos devem poder ser compilados nesse sistema utilizando o compilador GCC. A entrada deve sempre ser lida pela entrada padrão (geralmente do teclado) e a saída deve ser impressa na saída padrão.

6 Ranking

O código poderá ser enviado, ainda a ser divulgado, para avaliações diárias, onde todos os programas serão questionados para as mesmas entradas e serão ranqueados de acordo com o melhor resultado para a entrada.

As entradas utilizadas no sistema de *Ranking* não será divulgada.

O sistema de *Ranking* será automático e será executado em um computador com Linux, se o código não for compilado apenas será avisado que não foi compilado e nada mais.

7 Avaliação

Nesse trabalho diversos quesitos serão avaliados para geração da nota, sendo eles:

- Estrutura de Dados
  - Deverá ser utilizado a estrutura de listas ligadas dinamicamente para representar as estruturas de tamanho arbitrário;
  - Caso seja utilizado ordenação **não** utilizar o *Bubble Sort*.
- Criatividade
  - Desenvolver um programa não depende de uma receita de bolo pronto, mas sim da criatividade no momento de utilizar as estruturas de dados.
- Desempenho
  - A minimização do uso de buscas (ou garantir que ela ocorra mais rapidamente, como o uso de uma busca binária ao invés de uma sequencial) será levado em conta.
- Menor resultado
  - O programa deverá tentar imprimir a melhor sequencia de ações para minimizar os ataques, porém essa não é uma tarefa trivial e 10 pontos serão reservados para classificar os melhores resultados.
- Adequamento a especificação
  - Os programas devem seguir rigorosamente as especificações.