

Trabalho Prático 01 - Estruturas de Dados I (BCC202)

Marco Antonio M. Carvalho
Universidade Federal de Ouro Preto
Departamento de Computação

30 de setembro de 2021

Instruções

- Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado;
- Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas;
- A avaliação considerará o tempo de execução e o percentual de respostas corretas;
- Eventualmente realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;
- Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada;
- Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software;
- Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota e frequência;
- Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos;
- Não serão considerados algoritmos parcialmente implementados.

1 Alocação dinâmica de tipos abstratos de dados

Monstros de Duelo é o mais famoso campeonato de cartas do mundo. Cada duelo é disputado por dois jogadores, onde cada jogador começa com 8.000 Pontos de Vida e com o seu Deck (Baralho). Cada carta representa um monstro que tem os atributos de Ataque, Defesa e Habilidade. Estamos no duelo final com os dois maiores duelistas da história. De um lado Charlinho, um garoto que gosta muito de estudar, mas que também sabe sentir o coração das cartas. Do outro lado Gilmar, que é reconhecido como o Mestre das Cartas. É a vez de Charlinho, e ele acaba de combinar seus monstros na mesa para obter o proibido “Prassódia”.

Prassódia é o monstro mais forte do jogo, e quando invocado, a partida é finalizada na hora e o jogador que o invocou ganha o duelo. Para invocar Prassódia é necessário combinar duas ou mais cartas na mesa somando ataque com ataque, defesa com defesa e habilidade com habilidade para obter os atributos de Prassódia informados no início do duelo.

Se uma carta for escolhida para ser combinada, os atributos da carta tem que ser utilizados por inteiro, sendo inválido apenas utilizar o ataque ou apenas utilizar a defesa ou apenas a habilidade daquela carta e ainda não podendo utilizar apenas uma parte do ataque e/ou uma parte da defesa e/ou uma parte da habilidade.

Gilmar se espanta por um momento, pois ele foi o único em toda a história que conseguiu invocar Prassódia, mas acaba duvidando da jogada de Charlinho porque haviam várias cartas na mesa, o que tornaria tal jogada muito suspeita.

Você é o Juiz do duelo e ficou responsável por avaliar se o Charlinho realmente conseguiu invocar ou não Prassódia com as cartas na mesa.

Especificação da Entrada

A primeira linha da entrada contém 4 inteiros, o primeiro inteiro n representa o número de cartas na mesa seguido por 3 inteiros A , D e H que representam respectivamente o ataque, a defesa e a habilidade exata para invocar Prassódia. Cada uma das n linhas seguintes representam uma carta da mesa, onde cada linha contém os atributos na ordem X , Y , Z ($1 \leq X, Y, Z \leq 50$) ataque, defesa e habilidade respectivamente.

Especificação da Saída

A saída deverá conter “Y” caso Charlinho consiga realizar uma combinação válida para invocar Prassódia ou “N” caso contrário.

Exemplo de Entrada 1

```
2 5 20 10
8 10 7
9 10 3
```

Exemplo de Saída 1

N

Exemplo de Entrada 2

```
3 13 9 15
7 6 5
6 3 10
5 9 15
```

Exemplo de Saída 2

Y

Estrutura do código

O código-fonte deve ser modularizado corretamente em três arquivos: `principal.c`, `duelo.h` e `duelo.c`. O arquivo `principal.c` deve apenas invocar as funções e procedimentos definidos no arquivo `duelo.h`. A separação das operações em funções e procedimentos está a cargo do aluno, porém, não deve haver acúmulo de operações dentro uma mesma função/procedimento.

Além disto, a informação de cada carta deve ser armazenada em um tipo abstrato de dados criado especificamente para isso. Por fim, todas as informações das cartas da mesa devem ser armazenadas em um vetor alocado dinamicamente (e posteriormente liberado) para cada caso de teste.

Diretivas de Compilação

```
$ gcc duelo.c -c
$ gcc principal.c -c
$ gcc duelo.o principal.o -o programa
```