



**Universidade Federal do Espírito Santo**  
**Departamento de Informática**

**1ª exercício avaliativo (EA1)**  
**Programação II (INF16153) - UFES**

28 de maio de 2025

**Liga Online de jogadores de Estratégia**

Diversos jogos online utilizam medidas de desempenho para mostrar quão bom é um jogador. Considere um servidor de um jogo de estratégia em tempo real, em que dois jogadores disputam partidas entre si. Uma partida necessariamente terminará ou com a vitória de algum jogador ou com empate entre os jogadores. Em cada partida, os jogadores devem extrair recursos naturais do mapa e usá-los (ou não) durante a partida para derrotar o oponente. Caso isso não ocorra em determinado intervalo de tempo, a partida é considerada como empate.

Os servidores são limitados a até 100 jogadores e 1000 registros de partidas distintas entre eles. Com as partidas, o servidor atribui uma pontuação de liga (LP) a cada jogador, com base no seu desempenho em suas partidas. Cada partida possui um resultado e uma quantidade de recursos extraídos para cada jogador, que serão utilizados para cálculo na pontuação do jogador. Como é um jogo de estratégia, nem sempre o jogador que extraiu mais recursos vence a partida.

Todo jogador possui uma pontuação positiva, iniciando em valor zero, que será alterada a cada partida jogada. Independente do resultado, um valor  $\Delta P$  é somado a pontuação de liga do jogador. Vale ressaltar que, necessariamente,  $\Delta P > 0$  em caso de vitória,  $\Delta P < 0$  em caso de derrota. Em caso de empate, o sinal de  $\Delta P$  de cada jogador dependerá da quantidade de recursos extraída do mapa por cada jogador. Os valores de  $\Delta P$  são definidos a seguir ( $N_{jog}$  é a pontuação do jogador referido e  $N_{op}$  é a pontuação do jogador oponente):

1. Se jogador ganhou extraindo mais recursos que o oponente:  $\Delta P = N_{jog} - N_{op}$
2. Se jogador ganhou extraindo menos recursos que o oponente:  $\Delta P = 1,5 \times (N_{op} - N_{jog})$
3. Se jogador perdeu extraindo mais recursos que o oponente:  $\Delta P = 1,5 \times (N_{op} - N_{jog})$
4. Se jogador perdeu extraindo menos recursos que o oponente:  $\Delta P = N_{jog} - N_{op}$
5. Se empate, o jogador com mais recursos tem  $\Delta P = \frac{-|N_{op} - N_{jog}|}{2}$ , e o com menos recursos tem  $\Delta P = \frac{+|N_{op} - N_{jog}|}{2}$

Obs: A pontuação de liga não pode ser inferior à zero! Lembrando que  $\Delta P$  sempre é **somado** aos LPs (ele é negativo em caso de derrota ou empate desfavorável, e positivo em caso de vitória ou empate favorável)

Dependendo de cada pontuação, o jogador é colocado em uma das seguintes categorias:

- Até 200 LP: "ROOKIE"
- De 201 a 400 LP: "IRON"
- De 401 a 600 LP: "BRONZE"
- De 601 a 800 LP: "SILVER"
- De 801 a 1000 LP: "GOLD"
- De 1001 a 1500 LP: "PLATINUM"
- De 1501 a 2000 LP: "DIAMOND"
- Acima de 2000 LP: "MASTER"

Considere que cada partida é disputada por apenas dois jogadores. Cada partida é lida (uma por linha) com o seguinte formato:

ID_J1 ID_J2 ID_VENCEDOR NREC_J1 NREC_J2
---

- ID\_J1 e ID\_J2 são os IDs (códigos inteiros únicos que identificam cada jogador)
- ID\_VENCEDOR deve ser igual a ID\_J1 (caso J1 vença), ID\_J2 (caso J2 vença) ou -1 em caso de empate.
- NREC\_J1 e NREC\_J2 são as quantidades de recursos ( $n^\circ$  inteiro) extraídos de cada jogador.

Escreva um programa em C que leia os registros de múltiplas partidas de um servidor em ordem cronológica, e em seguida gere um relatório sobre alguns jogadores. Um exemplo de registro é mostrado a seguir:

1 2 1 370 70 (IDs dos jogadores, seguido do ID vencedor e quantid. de recursos extraídos)
1 3 3 140 120
2 3 -1 50 70
-1 (identificador de fim de registro)
3 ( $n^\circ$ de jogadores a ser exibida a liga)
1 2 4 (IDs dos jogadores que deve ser mostrada a liga)

Neste pequeno exemplo, temos 3 partidas: 1 vs 2, onde 1 venceu extraíndo mais recursos (ganhará 300 pontos, enquanto 2 “ganhará” -300 pontos); 1 vs 3, onde 3 venceu extraíndo menos recursos (ganhará  $1,5 \cdot (140 - 120) = 30$  pontos, enquanto J1 “ganhará” -30 pontos); e 2 vs 3, onde houve empate, e o jogador 2 ganhará 10 pontos (pois extraiu menos recursos e foi capaz de “segurar” o oponente com mais recursos), enquanto o jogador 3 perderá 10 pontos (pois extraiu mais recursos e não foi capaz de vencer). Isso quer dizer que J1 terá  $300 - 45 = 255$  pontos. Já J2 perderia 300 pela primeira partida, mas como ele começa com zero pontos e não

é permitida pontuação negativa, ele não perde nada. Em seguida, ele ganha 10 pontos pelo empate com J3. Já J3, ganhou 30 pontos e em seguida perde 10 pontos, totalizando 20 pontos.

O identificador de fim de registro (-1) indica que não haverá mais registros a serem lidos na entrada. Na sequência, será solicitado quais os IDs de jogadores deve ser exibido o desempenho. Se o ID existir no servidor, deve ser impresso na tela a liga do mesmo, no padrão:

```
Jogador ID: NUM_VIT/NUM_DERROTAS/NUM_EMPATES (NOME_DA_LIGA)
```

Se o ID não existir, ou seja, não existe registro do jogador no servidor em questão, deverá ser exibido:

```
Jogador ID: -
```

Sendo assim, para este exemplo, essa é a saída esperada:

```
Jogador 1: 1/1/0 (IRON)
Jogador 2: 0/1/1 (ROOKIE)
Jogador 4: -
```

#### Dicas:

- Um jogador pode jogar várias partidas dentro do servidor. Logo, o programa deve ser capaz de agregar as informações.
- Vocês tem acesso aos casos de teste e *script* de correção. Use-os com inteligência.

#### Regras gerais

- A atividade é **individual**. Todas as questões serão testadas e plágio não será tolerado
- Seu programa deve, obrigatoriamente, utilizar o *template* disponibilizado junto com o exercício. A correção é feita utilizando o *script* de correção (já conhecido)
- Seu programa deve simular o encapsulamento dos TADs
- Todos os valores máximos de arrays são fornecidos nos *templates*
- **Números de ponto flutuante deve ter precisão simples e apenas duas casas decimais devem ser impressas**