

MC202 — ESTRUTURAS DE DADOS

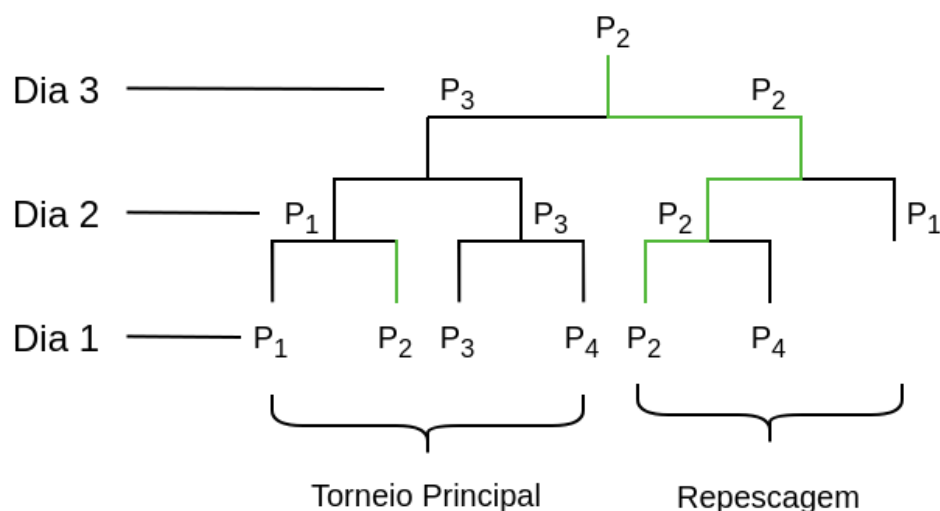
Lab 04 — Fila

Problema

Ocorrerá um torneio de xadrez com 2^N participantes, enumerados de 1 a 2^N . O primeiro participante P_1 enfrentará o segundo participante P_2 na primeira partida, P_3 enfrentará P_4 na segunda partida e assim por diante. O vencedor da partida entre P_1 e P_2 enfrentará o vencedor de P_3 e P_4 na primeira partida da segunda fase, o vencedor de P_5 e P_6 enfrentará o vencedor de P_7 e P_8 na segunda partida da segunda fase e assim por diante.

O torneio será realizado no formato de dupla eliminação, o que significa que são necessárias duas derrotas para que um participante seja eliminado. Ao ser derrotado pela primeira vez, o participante é enviado para a repescagem.

O formato da repescagem é o mesmo do torneio principal: o perdedor de P_1 e P_2 enfrenta o perdedor de P_3 e P_4 na primeira partida da repescagem e assim por diante. As fases da repescagem são intercaladas com as fases do torneio principal, ou seja, após a primeira rodada de partidas no torneio principal, a primeira rodada da repescagem é realizada. A cada dia, uma rodada do torneio principal será realizada pela manhã e uma rodada da repescagem será realizada a tarde. Uma rodada da repescagem conta com a participação dos vencedores da rodada da repescagem do dia anterior (caso não seja o primeiro dia) e os perdedores da manhã do respectivo dia, nessa ordem. O vencedor do torneio principal deve esperar que a repescagem termine para conhecer seu adversário na grande final (observe a figura a seguir).



O torneio da imagem corresponde ao exemplo de entrada. A linha verde indica o percurso do campeão (P_2) até o título.

Cada participante P_i tem a sua habilidade dada por H_i , que representa a média de atributos importantes no jogo, como raciocínio, concentração e memória. Quando dois participantes P_i e P_j disputam uma partida, o mais habilidoso vence e sua habilidade é decrementada da habilidade do adversário, devido ao desgaste da partida. Já o perdedor tem sua habilidade reduzida pela metade e é enviado para a repescagem. Antes de suas próximas partidas, os participantes têm tempo para descansar e conseguem recuperar sua habilidade em no máximo K , porém nunca ultrapassam seus valores iniciais (H_i e H_j). Na repescagem, as regras das partidas são as mesmas: o mais habilidoso vence e sua habilidade é decrementada da habilidade do adversário, porém o perdedor é desclassificado do torneio. Se dois participantes possuem o mesmo nível de habilidade, o de menor índice vence. Antes da grande final os finalistas terão alguns dias para descansar e jogarão em seu mais alto nível, ou seja, com habilidade inicial.

Dado o número de participantes e o nível de habilidade inicial de cada um, determine quem vencerá o torneio.

Entrada

A primeira linha é composta por dois inteiros N ($2 \leq N \leq 16$) e K ($0 \leq K \leq 1000$). A próxima linha contém 2^N inteiros H_i ($1 \leq H_i \leq 1000$) indicando a habilidade do i -ésimo participante do torneio, para $i = 1 \dots 2^N$.

Saída

Para cada partida do torneio principal (exceto a final), o seu programa deverá imprimir uma linha com o número da partida (começando de 1), o identificador dos participantes e o vencedor, em ordem de menor para o maior, seguindo o formato "Partida T : P_i vs P_j venceu P_i ".

Para cada partida da repescagem, o seu programa deverá imprimir uma linha com "Partida L da repescagem: P_i vs P_j venceu P_i ", em que L é o contador de partidas na repescagem (começando de 1) e é diferente do contador de partidas do torneio principal e P_i e P_j são os identificadores dos participantes em ordem crescente.

Para a grande final seu programa deverá imprimir uma linha com "Final do torneio: P_i vs P_j campeão P_j ", em que P_i e P_j são os finalistas, em ordem de identificador.

Exemplo

Entrada

2 20

100 90 79 37

Saída

```
Partida 1: 1 vs 2 venceu 1
Partida 2: 3 vs 4 venceu 3
Partida 1 da repescagem: 2 vs 4 venceu 2
Partida 3: 1 vs 3 venceu 3
Partida 2 da repescagem: 1 vs 2 venceu 2
Final do torneio: 2 vs 3 campeao 2
```

Observações

- Neste laboratório é obrigatório o uso da estrutura de dados Fila.
- Você deverá submeter os seguintes arquivos no susy: **lab04.c**, **fila.h** e **fila.c**.
- Para as turmas A, B e C, esse laboratório tem peso 2.
- Observações sobre SuSy:
 - Versão do GCC: 4.4.7 20120313 (Red Hat 4.4.7-17).
 - *Flags* de compilação: `-g -ansi -pedantic-errors -Wall -Werror -lm`
 - Utilize comentários do tipo `/* comentário */;`
comentários do tipo `//` serão tratados como erros pelo SuSy
 - Tempo máximo de execução: 1 segundo.
- Além das observações acima, esse laboratório será avaliado pelos critérios:
 - Indentação de código.
 - Organização.
 - Corretude.