

# Problema E

## Libertadores

Arquivo fonte: libertadores.c, libertadores.cpp ou libertadores.java

A Copa Libertadores da América é a principal competição de futebol entre clubes profissionais da América do Sul, organizada pela Confederação Sul-Americana de Futebol (CONMEBOL). Ela é conhecida por ter um regulamento muito complicado, principalmente nas fases das oitavas, quartas e semi-final.

Nessas fases são jogadas partidas de ida e volta no sistema mata-mata. Ganha quem fizer a maior pontuação no acumulado das duas partidas, sendo 3 pontos para vitória e 1 ponto em caso de empate, ambos por partida. Em caso de igualdade na pontuação, são critérios de desempate:

- 1) saldo de gols (número de gols a favor menos o número de gols contra).
- 2) mais gols marcados na casa do adversário.
- 3) disputa por pênaltis.

Todos os critérios devem ser aplicados considerando o acumulado das duas partidas.

Será que você consegue elaborar um algoritmo que, dados os resultados das partidas de ida e de volta, ele identifica o time vencedor?

### Entrada

A primeira linha de entrada indica o número de casos de teste  $N$  ( $1 \leq N \leq 100$ ). Cada caso de teste é composto por dois placares: o resultado da partida 1 e o resultado da partida 2. O placar é representado pelo formato  $M \times V$ , onde  $M$  ( $1 \leq M \leq 100$ ) é o número de gols do time mandante da partida e  $V$  ( $1 \leq V \leq 100$ ) é o número de gols do time visitante. Como em cada caso de teste existem 2 partidas, considere que o Time 1 é sempre o mandante da primeira e o visitante da segunda e vice-versa para o Time 2.

### Saída

Para cada caso de teste, imprima uma linha contendo "Time 1" (sem aspas) caso o Time 1 seja o vencedor do mata-mata, "Time 2" (sem aspas) caso o Time 2 seja o vencedor do mata-mata e "Penaltis" (sem aspas) caso não seja possível identificar o vencedor no tempo convencional.

### Exemplo de entrada

```
4
1 x 1
2 x 1
2 x 0
2 x 1
1 x 1
2 x 2
3 x 1
3 x 1
```

### Exemplo de saída

```
Time 2
Time 1
Time 1
Penaltis
```