Problema M

ScoreBoard - Final

Arquivo fonte: scoreboard.{ c | cpp | java | py }

Autor: Prof. Me. Sérgio Luiz Banin (Fatec São Paulo)

Na primeira etapa da Maratona InterFatecs, vocês tiveram que criar um programa para classificar os times para a final. Esse desafio gerou algumas dúvidas, principalmente sobre como a pontuação de cada equipe é calculada.

A quantidade de problemas resolvidos é um critério claro, mas o fato é que muitos times acertam a mesma quantidade. Por isso, é preciso de um segundo critério: o tempo. Esse critério é simples, mas nem sempre fica claro como é feito o cálculo.

Quando um time acerta um problema, registra-se o minuto em que isso acontece. Depois, somam-se todos esses minutos, que vão formar o tempo total do time. A esse tempo, adicionam-se 20 minutos de multa para cada tentativa errada, mas só se o time tiver conseguido resolver o problema. Para os problemas não resolvidos, as tentativas erradas não contam.

Veja este exemplo em que um time resolveu 4 problemas:

	Tentativas	Tempo	Resolvido	Pontuação
Problema A	1	30	Sim	30
Problema B	0	0	Não	0
Problema C	3	0	Não	0
Problema D	3	210	Sim	210 + 40
Problema E	1	70	Sim	70
Problema F	2	150	Sim	150 + 20
Totais			**4**	**520**

Figura M.1: Apuração dos pontos de um time

Para definir a classificação dos times na Maratona são usados esses dois parâmetros: a quantidade de problemas em ordem decrescente e o tempo em ordem crescente. O campeão será o time que resolver o maior número de problemas no menor tempo.

Muito bem, agora que você já entendeu o processo, deve escrever um programa que leia os dados envolvidos e produza a classificação dos times.

Entrada

A entrada contém um único caso de teste. Na primeira linha há um número inteiro NP que representa o número de problemas da Maratona $(4 \leq NP \leq 20)$. Na segunda linha há um número inteiro que representa a quantidade de times QT participantes da Maratona $(6 \leq QT \leq 1000)$.

Em seguida há QT pares de linhas. A primeira linha de cada par contém dois strings de no máximo 100 caracteres cada: o nome do time e o nome da Fatec, separados pelo caractere pipe ". A segunda linha contém NP pares de inteiros representando a solução dos problemas, sendo que o primeiro inteiro é o número de tentativas e o segundo é o tempo da solução correta. Se este segundo inteiro for zero o problema não foi resolvido.

Saída

A saída deve conter uma linha para cada time presente na entrada. Cada linha precisa exibir quatro informações: o nome do time, o nome da Fatec, o número de problemas resolvidos e o tempo total. O formato da saída deve ser seguir literalmente o exemplo mostrado. Entre o nome do time e o da Fatec deve haver um hífen com um espaço em branco antes e depois dele. Após o nome da Fatec deve haver um espaço em branco e em seguida entre parênteses devem estar o número de problemas e o tempo.

As linhas devem estar ordenadas do melhor para o pior time. O primeiro critério de ordenação é a quantidade de problemas resolvidos em ordem decrescente. Havendo empate nesse critério o desempate deve ser feito usando o tempo em ordem crescente.

Em caso muito remoto de dois times empatarem segundo os dois critérios, a ordem deve ser alfabética pelo nome do time.

Exemplo de Entrada 1	**Exemplo de Saída 1**
6 8 Time 01|Fatec A 0 0 1 0 4 0 1 0 3 170 2 0 Time 02|Fatec A 0 0 1 90 1 50 0 0 1 180 0 0 Time 03|Fatec B 0 0 2 80 0 0 1 50 2 110 2 170 Time 04|Fatec B 4 150 1 20 2 70 2 0 1 40 2 110 Time 05|Fatec B 0 0 1 60 1 30 0 0 1 160 1 100 Time 06|Fatec C 0 0 1 120 0 0 0 0 3 320 0 0 Time 07|Fatec C 0 0 1 50 1 80 0 0 1 120 0 0 Time 08|Fatec D 0 0 1 40 2 140 0 0 3 80 0 0	Time 04 - Fatec B (5,490) Time 05 - Fatec B (4,350) Time 03 - Fatec B (4,470) Time 07 - Fatec C (3,250) Time 02 - Fatec A (3,320) Time 08 - Fatec D (3,320) Time 06 - Fatec C (2,480) Time 01 - Fatec A (1,210)