

Problema D. Copa América 2019

Arquivo-fonte: `copa.c` ou `copa.cpp`

A Copa América 2019 (oficialmente CONMEBOL Copa América Brasil 2019) começou há uma semana. No jogo de abertura, o anfitrião Brasil venceu a Bolívia por 3×0 . No dia seguinte, a Colômbia ganhou da Argentina por 2×0 .

Um amigo seu quer aproveitar a Copa América para ganhar dinheiro com um aplicativo para *smartphone* que mostra a tabela atualizada para os usuários. Para isso, ele criou um sistema que se conecta diretamente ao site da CONMEBOL e faz o *download* dos resultados dos jogos já realizados. O problema é que ele se complicou na hora de fazer este sistema interpretar os resultados e calcular o número de pontos de cada seleção. Por isso, ele pediu a sua ajuda para implementar essa funcionalidade no aplicativo.

Você deve fazer um programa que leia a lista de resultados dos jogos já realizados e calcule o número de pontos das quatro seleções que estão em um mesmo grupo.



Observações

Uma seleção ganha 3 pontos para cada vitória e 1 ponto para cada empate. Derrotas não alteram sua pontuação.

Entrada

A entrada é dividida em duas partes, separadas por uma linha em branco.

A primeira parte são os dados lidos diretamente do site da CONMEBOL. Esta parte começa com um inteiro N indicando o número de partidas já realizadas. Cada uma das N linhas seguintes contém o resultado de uma partida, no seguinte formato:

`time1 placar1 x placar2 time2`

Onde:

- `time1` e `time2` são os nomes das seleções (sequências de caracteres sem espaços);
- `placar1` é o número de gols feitos pelo `time1`;
- `placar2` é o número de gols feitos pelo `time2`;

Observações

Para armazenar estes dados, você deve obrigatoriamente utilizar um vetor “partida” como definido abaixo:

```
struct TipoPartida {  
    char time1[31], time2[31];  
    int placar1, placar2;  
};
```

```
TipoPartida partida[50];
```

A segunda parte da entrada indica quais seleções estão no grupo para o qual o aplicativo está calculando a classificação. Portanto, esta parte é composta por exatamente 4 linhas, cada uma delas indicando o nome de uma seleção. Note que é possível que uma ou mais destas seleções ainda não tenham realizado nenhuma partida.

Restrições: $N \leq 50$; $0 \leq \text{placar1}, \text{placar2} \leq 10$; todos os nomes de seleções contêm no máximo 30 caracteres e não contêm espaços.

Saída

Seu programa deve gerar 4 linhas de saída, cada uma contendo o nome de uma das seleções seguida pelo número de pontos conquistados por ela até o momento. A ordem de saída deve ser a mesma de entrada, ou seja, as seleções não devem ser reordenadas de acordo com a pontuação.

Exemplos

Entrada	Saída
3 Brasil 3 x 0 Bolivia Venezuela 0 x 0 Peru Bolivia 1 x 3 Peru Brasil Bolivia Venezuela Peru	Brasil 3 Bolivia 0 Venezuela 1 Peru 4
10 Brasil 3 x 0 Bolivia Venezuela 0 x 0 Peru Argentina 0 x 2 Colombia Paraguai 2 x 2 Catar Uruguai 4 x 0 Equador Japao 0 x 4 Chile Bolivia 1 x 3 Peru Brasil 0 x 0 Venezuela Colombia 1 x 0 Catar Argentina 1 x 1 Paraguai Argentina Colombia Paraguai Catar	Argentina 1 Colombia 6 Paraguai 2 Catar 1

Dica: para não ter que digitar a entrada toda vez que for testar seu programa, crie um arquivo contendo cada entrada e execute seu programa fazendo redirecionamento de entrada:

```
./a.out < entrada1.txt
```

Assim, toda leitura feita da entrada padrão (`scanf` e `cin`) lerá dados do arquivo `entrada1.txt`. Você não precisará digitar a entrada novamente e ela não aparecerá na tela, apenas o resultado do programa.