

## F. Rinha de Pokemon

time limit per test: 1 second

memory limit per test: 256 megabytes



Rinha é um esporte proibido em muitos lugares, mas não em Kanto, onde os próprios pokemons podem decidir se querem participar ou não da atividade.

A rinha de pokemons funciona de uma maneira bem peculiar. Cada pokemon utiliza apenas o seu status de força para lutar. Assim, os pokemons se cadastram em um torneio e as batalhas ocorrem em duelos (PVP). De todos os pokemons participantes, lutam primeiro os dois pokemons mais fortes. Com o resultado da luta, o pokemon perdedor é eliminado do torneio e o vencedor avança para a próxima fase, perdendo em força o valor da força do pokemon perdedor. Caso ocorra uma batalha onde os dois pokemons possuam a mesma força, ambos são eliminados. Assim, a batalha seguinte recebe os status de todos os pokemons restante atualizados e a próxima batalha ocorre entre os dois pokemons mais fortes daquele instante, até que reste apenas um pokemon.

Você foi contratado pelo ginásio de IDPeridian para simular o torneio de rinha de pokemons. Dada a lista de pokemons participantes e a força de cada um, você deverá escrever um programa para organizar todas as batalhas e determinar o vencedor do torneio, se houver um.

### Input

A entrada contém um único caso de teste. A primeira linha contém um inteiro  $P$  ( $1 \leq P \leq 2 * 10^4$ ) que indica o número de pokemons que irão participar do torneio. As próximas  $P$  linhas contêm uma string  $S_i$  ( $1 \leq |S_i| \leq 25$ ) e um inteiro  $F$  ( $1 \leq F \leq 10^5$ ), separados por um espaço, que indicam o nome do pokemon e a sua respectiva força.

### Output

A saída irá conter uma ou mais linhas descrevendo cada uma das batalhas que ocorreram. Cada linha deverá conter o nome dos pokemons, suas respectivas forças e o nome do vencedor, conforme os exemplos. A última linha deverá conter a frase "S venceu com F", onde  $S$  é o nome do pokemon vencedor e  $F$  é a sua força restante, caso exista um vencedor, ou "nenhum vencedor", caso contrário.

### Examples

#### input

```
4
Pikachu 55
Charizard 88
Gengar 60
Muk 40
```

#### output

```
Charizard (88) x (60) Gengar : Charizard venceu
Pikachu (55) x (40) Muk : Pikachu venceu
Charizard (28) x (15) Pikachu : Charizard venceu
Charizard venceu com 13
```

#### input

```
6
Cubone 60
Pikachu 35
Magmar 27
Abra 25
Metapod 0
```

Magmar 28

[Copy](#) [Copy](#) [Copy](#)

**output**

[Copy](#)

Cubone (60) x (35) Pikachu : Cubone venceu  
Magmar (28) x (27) Magmar : Magmar venceu  
Cubone (25) x (25) Abra : empate  
Magmar (1) x (0) Metapod : Magmar venceu  
Magmar venceu com 1

**input**

[Copy](#)

4  
Arbok 44  
Articuno 87  
Pikachu 44  
Ratata 1

**output**

[Copy](#)

Articuno (87) x (44) Pikachu : Articuno venceu  
Arbok (44) x (43) Articuno : Arbok venceu  
Ratata (1) x (1) Arbok : empate  
nenhum vencedor

**IDP - TAA - 2025/02**

**Private**

Participant



→ **About Group**



Este grupo tem o objetivo de organizar as atividades de programação da disciplina de Técnicas de Programação e Análise de Algoritmos.

[Group website](#)

→ **Group Contests**

- TAA - LEA 05
- TAA - LEE 05
- TAA - LEA 04
- TAA - LEE 04
- TAA - AS 01
- TAA - LEA 03
- TAA - LEE 03
- TAA - LEA 02
- TAA - LEE 02
- TAA - LEA 01
- TAA - LEE 01
- ET - Exercício de Testes

**TAA - LEE 05**

**Finished**

Contestant



→ **Last submissions**

Submission	Time	Verdict
<a href="#">347153488</a>	Nov/02/2025 22:43	Accepted

Supported by

