B. Order Book

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In this task you need to process a set of stock exchange orders and use them to create *order book*.

An *order* is an instruction of some participant to buy or sell stocks on stock exchange. The order number i has price p_i, direction d_i — buy or sell, and integer q_i. This means that the participant is ready to buy or sell q_i stocks at price p_i for one stock. A value q_i is also known as a *volume* of an order.

All orders with the same price p and direction d are merged into one *aggregated* order with price p and direction d. The volume of such order is a sum of volumes of the initial orders.

An order book is a list of aggregated orders, the first part of which contains sell orders sorted by price in descending order, the second contains buy orders also sorted by price in descending order.

An order book of depth s contains s best aggregated orders for each direction. A buy order is better if it has higher price and a sell order is better if it has lower price. If there are less than s aggregated orders for some direction then all of them will be in the final order book.

You are given n stock exhange orders. Your task is to print order book of depth s for these orders.

Input

The input starts with two positive integers n and s ($1 \le n \le 1000, 1 \le s \le 50$), the number of orders and the book depth.

Next n lines contains a letter d_i (either 'B' or 'S'), an integer p_i ($0 \le p_i \le 10^5$) and an integer q_i ($1 \le q_i \le 10^4$) — direction, price and volume respectively. The letter 'B' means buy, 'S' means sell. The price of any sell order is higher than the price of any buy order.

Output

Print no more than $2s$ lines with aggregated orders from order book of depth s. The output format for orders should be the same as in input.

Examples

input

```
6 2
B 10 3
S 50 2
S 40 1
S 50 6
B 20 4
B 25 10
```

output

```
S 50 8
S 40 1
B 25 10
B 20 4
```

Note

Denote *(x, y)* an order with price x and volume y. There are 3 aggregated buy orders (10, 3), (20, 4), (25, 10) and two sell orders (50, 8), (40, 1) in the sample.

You need to print no more than two best orders for each direction, so you shouldn't print the order (10 3) having the worst price among buy orders.