

## COIN COLLECTOR

In a certain country, there are  $N$  denominations of coins in circulation, including the 1-cent coin. Additionally, there's a bill whose value of  $K$  cents is known to exceed any of the coins. There's a coin collector who wants to collect a specimen of each denomination of coins. He already has a few coins at home, but currently he only carries one  $K$ -cent bill in his wallet. He's in a shop where there are items sold at all prices less than  $K$  cents (1 cent, 2 cents, 3 cents,  $\dots$ ,  $K - 1$  cents). In this shop, the change is given using the following algorithm:

1. Let the amount of change to give be  $A$  cents.
2. Find the highest denomination that does not exceed  $A$ . (Let it be the  $B$ -cent coin.)
3. Give the customer a  $B$ -cent coin and reduce  $A$  by  $B$ .
4. If  $A = 0$ , then end; otherwise return to step 2.

The coin collector buys one item, paying with his  $K$ -cent bill.

Your task is to write a program that determines:

- How many different coins that the collector does not yet have in his collection can he acquire with this transaction?
- What is the most expensive item the store can sell him in the process?

### INPUT

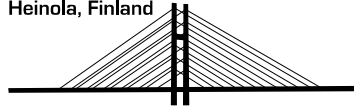
standard input

The input is read from ~~a text file named coins.in.~~ The first line of the input file contains the integers  $N$  ( $1 \leq N \leq 500\,000$ ) and  $K$  ( $2 \leq K \leq 1\,000\,000\,000$ ). The following  $N$  lines describe the coins in circulation. The  $(i + 1)$ -th line contains the integers  $c_i$  ( $1 \leq c_i < K$ ) and  $d_i$ , where  $c_i$  is the value (in cents) of the coin, and  $d_i$  is 1, if the collector already has this coin, or 0, if he does not. The coins are given in the increasing order of values, that is,  $c_1 < c_2 < \dots < c_N$ . The first coin is known to be the 1-cent coin, that is,  $c_1 = 1$ .

### OUTPUT

standard output

The output is written into ~~a text file named coins.out.~~ The first line of the output file should contain a single integer — the maximal number of denominations that the collector does not yet have, but could acquire with a single purchase. The second line of the output file should also contain a single integer — the maximal price of the item to buy so that the change given would include the maximal number of new denominations, as declared on the first line.



**EXAMPLE**

standard input

~~coins.in~~

```
7 25
1 0
2 0
3 1
5 0
10 0
13 0
20 0
```

standard output

~~coins.out~~

```
3
6
```