Cake

It's your friend Sharon's birthday. You decided to give her a large piece of chocolate cake for her. Sharon loves to eat chocolate cake. "Yumm!", she exclaims, when she sees the large chocolate cake that you gave her. "Thanks a lot! I can't wait to eat the whole cake!", she says jubilantly.

However, there is one problem. The cake contains raisins. Sharon despises raisins. "I want to eat the chocolate cake, but I don't like raisins.", she says. However, her love for chocolate beats her hate of raisins. Nevertheless, she still has a limit as to how many blocks of raisins she would take.

The cake is of size 1 x N and consists of N blocks. Each block is of size 1 x 1. Each block can either contain no raisins or some amount of raisins. Each block can also contain different units of chocolate. A sample cake can be seen in the following diagram:

10	2	3	4	5

The cake above is of size 1 x 6, and has 3 blocks containing raisins (the purple-colored blocks). Each block (whether they contain raisins or not) contains a certain amount of chocolate. The first block (on the left) contains 10 units of chocolate and the next one contains 2 units of chocolate, and so on.

Sharon only wants a sequence of contiguous chocolate blocks since she wants to eat them in one go. Your job is to tell her the maximum amount of chocolate she can take while still not exceeding the maximum number of blocks containing raisins.

Input

The first line of input consists of two integers **N (1 <= N <= 100000) and K (1 <= K <= 30000)**, separated by a single space, denoting the length of the cake and the maximum number of blocks with raisins that Sharon can still tolerate. **N** rows follow. Each line consists of a character **T** and an integer **X (0 <= X <= 1000)**, separated by a single space, denoting whether the block contains raisins or not and the amount of chocolate in that block of cake. The character T can be either "R" (denoting that there are raisins in that block) or "C" (denoting that there are no raisins in that block).

Output

Print the maximum amount of chocolate (in units) that Sharon can eat without exceeding the maximum number of blocks containing raisins.

Sample Input 1

```
5 2
R 10
C 2
R 3
C 4
R 5
```

Sample Output 1

```
19
```

Sample Input 2

```
5 3
R 10
C 2
R 3
C 4
R 5
```

Sample Output 2

```
24
```

Explanation

In the first input, the cake is of size 5 (as illustrated in the diagram in the statement). This time, Sharon can take at most 2 blocks of cake containing raisins. The most amount of chocolate is achieved if Sharon takes the first four blocks, getting 19 units of chocolate in total.

In the second input, the cake remains the same. However, Sharon now tolerates up to 3 blocks of cake containing raisins. Therefore, Sharon can eat the whole cake, getting 24 units of chocolate in total, while still not exceeding the maximum number of blocks containing raisins that she can tolerate.

Skeleton

You are given the skeleton file `Cake.java`

Notes

1. You must either use **stack** or **queue** to solve this problem, whichever is suitable.
2. Your program might give out the correct answer, but killed on CodeCrunch. It means that your program is not efficient enough (i.e. it runs too slow). You should design a more efficient algorithm to solve this problem if this is the case. Consider the above note as a hint.