

Wall

This question is graded for 1%!

Statement

Fluffy the Hamster is building a wall by stacking bricks of the same size together. This wall consists of columns of bricks on the x -axis, where there is one column on each integer x -coordinate. The columns may have different heights. The height of a column is the number of bricks in it.

Fluffy the Hamster builds the wall as follows. Initially there are no bricks in any column. Fluffy starts from the column at $x = 0$, and builds the wall according to a blueprint. The blueprint is a series of N instructions of two types:

- **LEFT c** : Fluffy should move c columns leftwards. If Fluffy is currently at column x , then for each of the columns numbered $x - c$ to $x - 1$ (inclusive), Fluffy adds one additional brick to that column. After this instruction, Fluffy will be positioned at column $x - c$.
- **RIGHT c** : Fluffy should move c columns rightwards. If Fluffy is currently at column x , then for each of the columns numbered x to $x + c - 1$ (inclusive), Fluffy adds one additional brick to that column. After this instruction, Fluffy will be positioned at column $x + c$.

Help Fluffy compute how many columns have height at least H after the wall is constructed.

Constraints

- $1 \leq N \leq 10^5$
- $c > 0$
- Fluffy will not move beyond $x = 10^9$ and $x = -10^9$.

Input

The first line of input will contain two integers N and H .

The next N lines of input will each contain one operation as defined above.

Output

Print a single integer, the number of columns with height at least H after all the operations have been executed.

Important Note

For the purposes of this lab, you may only use the following data structures:

- Java built-in arrays
- Java API `ArrayList`,
- Java API `Stack`,
- Java API `LinkedList`,
- Java API `Pair`,
- Your own self-implemented versions of stack, queue, linked list and pair, including those from lecture.

You may also use any algorithms available in the Java API, e.g. `Arrays.sort`, `Collections.sort` etc.

All other data structures are **forbidden**. **Submissions which violate this rule will be given a score of 0.**

Examples

Sample Input	Expected Output
<pre>10 3 RIGHT 4 RIGHT 1 LEFT 3 RIGHT 2 LEFT 5 LEFT 5 LEFT 1 LEFT 5 RIGHT 1 LEFT 5</pre>	<pre>3</pre>

Notes

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided.

Skeleton File

You are given the skeleton file `Wall.java`. You should see the following contents when you open the file:

```
/**
 * Name      :
 * Matric. No :
 */

import java.util.*;

public class Wall {
    private void run() {

    }

    public static void main(String args[]) {
        Wall runner = new Wall();
        runner.run();
    }
}
```