Halves

Time limit: 0.5s **Memory limit:** 512M

You are given a grid of N integers, where N is a power of two. Rows and columns are numbered starting at 1 from top to bottom and left to right, respectively.

The grid initially consists of a single row containing the integers from 1 to N, in order. You are to handle Q updates and queries on the grid:

1. `X 0` - Cut the grid into left and right halves and put the left half on top of the right half.
2. `X 1` - Cut the grid into left and right halves and put the right half on top of the left half.
3. `Y 0` - Cut the grid into top and bottom halves and put the top half on the left of the bottom half.
4. `Y 1` - Cut the grid into top and bottom halves and put the bottom half on the left of the top half.
5. `Q x` - Determine the current row and column number of the value x.

For all of the cut operations, it is guaranteed that there will be at least two rows/columns in the direction which must be cut in half. Rows and columns are renumbered starting at 1 from top to bottom and left to right after each cut.

Constraints

$2 \leq N \leq 2^{30}$

$1 \leq Q \leq 5 \times 10^5$

$1 \leq x \leq N$

N is a power of two.

Subtask $\frac{1}{}$ [1/2 points]

$x = 1$

Subtask $\frac{}{2}$ [1/2 points]

No additional constraints.

Input Specification

The first line contains two space-separated integers, N and Q, the number of integers in the grid and the number of updates and queries, respectively.

The next Q lines each contain a query or update in one of the following formats:

1. `X 0` - Cut the grid into left and right halves and put the left half on top of the right half.
2. `X 1` - Cut the grid into left and right halves and put the right half on top of the left half.
3. `Y 0` - Cut the grid into top and bottom halves and put the top half on the left of the bottom half.

4. `Y 1` - Cut the grid into top and bottom halves and put the bottom half on the left of the top half.
5. `Q x` - Determine the current row and column number of the value x.

For all of the cut operations, it is guarenteed that there will be at least two rows/columns in the direction which must be cut in half.

Output Specification

For every query operation (type `Q`), output a line containing two space-separated integers, the current row and column number of the queried value.

Sample Input

```
8 9
Q 3
X 0
X 1
Q 1
Y 0
Q 8
Q 7
Y 1
Q 4
```

Sample Output

```
1 3
3 1
2 2
2 1
1 6
```

Explanation for Sample

The grid initially looks like this:

```
1 2 3 4 5 6 7 8
```

The first operation, `Q 3`, queries the current position of the value 3, which is row 1, column 3. Thus, the correct output for this query is `1 3`.

The second operation, `X 0`, cuts the grid into left and right halves and puts the left half on top of the right half:

```
1 2 3 4
5 6 7 8
```

The third operation, `X 1`, cuts the grid into left and right halves and puts the right half on top of the left half:

```
3 4
7 8
1 2
5 6
```

The fourth operation, `Q 1`, queries the current position of the value 1, which is row 3, column 1. Thus, the correct output for this query is `3 1`.

The fifth operation, `Y 0`, cuts the grid into top and bottom halves and puts the top half on the left of the bottom half:

```
3 4 1 2
7 8 5 6
```

The sixth operation, `Q 8`, queries the current position of the value 8, which is row 2, column 2. Thus, the correct output for this query is `2 2`.

The seventh operation, `Q 7`, queries the current position of the vallue 7, which is row 2, column 1. Thus, the correct output for this query is `2 1`.

The eighth operation, `Y 1`, cuts the grid into top and bottom halves and puts the bottom half on the left of the top half:

```
7 8 5 6 3 4 1 2
```

The ninth operation, `Q 4`, queries the current position of the value 4, which is row 1, column 6. Thus, the correct output for this query is `1 6`.