Java BitSet

Java's BitSet class implements a vector of bit values (i.e.: $false$ (0) or $true$ (1)) that grows as needed, allowing us to easily manipulate bits while optimizing space (when compared to other collections). Any element having a bit value of 1 is called a *set bit*.

Given 2 BitSets, B_1 and B_2, of size N where all bits in both BitSets are initialized to 0, perform a series of M operations. After each operation, print the number of *set bits* in the respective BitSets as two space-separated integers on a new line.

Input Format

The first line contains 2 space-separated integers, N (the length of both BitSets B_1 and B_2) and M (the number of operations to perform), respectively.
The M subsequent lines each contain an operation in one of the following forms:

- AND $<\text{set}> <\text{set}>$
- OR $<\text{set}> <\text{set}>$
- XOR $<\text{set}> <\text{set}>$
- FLIP $<\text{set}> <\text{index}>$
- SET $<\text{set}> <\text{index}>$

In the list above, $<\text{set}>$ is the integer 1 or 2, where 1 denotes B_1 and 2 denotes B_2.
$<\text{index}>$ is an integer denoting a bit's index in the BitSet corresponding to $<\text{set}>$.

For the binary operations AND, OR, and XOR, operands are read from left to right and the BitSet resulting from the operation replaces the contents of the *first operand*. For example:

```
AND 2 1
```

B_2 is the left operand, and B_1 is the right operand. This operation should assign the result of $B_2 \wedge B_1$ to B_2.

Constraints

- $1 \leq N \leq 1000$
- $1 \leq M \leq 10000$

Output Format

After each operation, print the respective number of *set bits* in BitSet B_1 and BitSet B_2 as 2 space-separated integers on a new line.

Sample Input

```
5 4
AND 1 2
SET 1 4
FLIP 2 2
OR 2 1
```

Sample Output

```
0 0
1 0
1 1
1 2
```

Explanation

Initially: $N = 5$, $M = 4$, $B_1 = \{0, 0, 0, 0, 0\}$, and $B_2 = \{0, 0, 0, 0, 0\}$. At each step, we print the respective number of *set bits* in B_1 and B_2 as a pair of space-separated integers on a new line.

$M_0 = AND\ 1\ 2$
$B_1 = B_1 \wedge B_2 = \{0, 0, 0, 0, 0\} \wedge \{0, 0, 0, 0, 0\} = \{0, 0, 0, 0, 0\}$
$B_1 = \{0, 0, 0, 0, 0\}$, $B_2 = \{0, 0, 0, 0, 0\}$
The number of *set bits* in B_1 and B_2 is 0.

$M_1 = SET\ 1\ 4$
Set $B_1[4]$ to $true$ (1).
$B_1 = \{0, 0, 0, 0, 1\}$, $B_2 = \{0, 0, 0, 0, 0\}$.
The number of *set bits* in B_1 is 1 and B_2 is 0.

$M_2 = FLIP\ 2\ 2$
Flip $B_2[2]$ from $false$ (0) to $true$ (1).
$B_1 = \{0, 0, 0, 0, 1\}$, $B_2 = \{0, 0, 1, 0, 0\}$.
The number of *set bits* in B_1 is 1 and B_2 is 1.

$M_3 = OR\ 2\ 1$
$B_2 = B_2 \vee B_1 = \{0, 0, 1, 0, 0\} \vee \{0, 0, 0, 0, 1\} = \{0, 0, 1, 0, 1\}$.
$B_1 = \{0, 0, 0, 0, 1\}$, $B_2 = \{0, 0, 1, 0, 1\}$.
The number of *set bits* in B_1 is 1 and B_2 is 2.