# 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 <set><set>
- OR  $<\!\!\operatorname{set}\!> <\!\!\operatorname{set}\!>$
- XOR < set > < set >
- FLIP <set> <index>
- SET <set> <index>

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

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

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

## **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

#### Sample Output

#### 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.

$$\begin{split} M_0 &= AND \ 1 \ 2 \\ B_1 &= B_1 \wedge B_2 = \{0,0,0,0,0\} \wedge \{0,0,0,0,0\} = \{0,0,0,0,0\} \\ B_1 &= \{0,0,0,0,0\}, \ B_2 = \{0,0,0,0,0\} \\ \text{The number of } set \ bits \ \text{in } B_1 \ \text{and } B_2 \ \text{is } 0. \end{split}$$

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

$$\begin{split} M_3 &= OR \ 2 \ 1 \\ B_2 &= B_2 \lor B_1 = \{0,0,1,0,0\} \lor \{0,0,0,0,1\} = \{0,0,1,0,1\}. \\ B_1 &= \{0,0,0,0,1\}, \ B_2 = \{0,0,1,0,1\}. \\ \text{The number of } set \ bits \ \text{in } B_1 \ \text{is } 1 \ \text{and } B_2 \ \text{is } 2. \end{split}$$