Hello World!

"Hello World" is a typical phrase used to indicate a beginner program written in a particular language to introduce it to new learners. However, do not let the problem title fool you. This is not a simple "print Hello World" problem.

This problem will introduce you to three different input-parsing methods that will be used during the course of the semester. It is important that you are familiar with the three different input-parsing methods. These methods will be used in the labs including sit-in labs, take-home labs and PE for CS1020.

Of course, there are more than three methods to parse inputs. The three given to you are just simple examples that you will encounter in this module. In this problem, you will encounter three basic techniques of parsing input, which includes the following methods:

1. Given an integer N, you should read N lines, each containing some information.
2. Read until you encounter a special character (e.g. read until -1 or 0).
3. Read until the very end of the file (i.e. read all lines in the input file).

In order to exercise the three different input-parsing methods, we first need to have an actual "problem" to solve. This is a simple logic problem. Given a query consisting of two bits (either "1" or "0") and a logical operator ("AND" or "OR"), output the result of the operation between the two given bits.

Input
The first line of input consists of a single integer, which can be either "1", "2", and "3". This integer represents the method of parsing inputs that you need to use, corresponding to the methods explained in the problem description above. For this problem, we use the special character "0" for method 2. Each query will have a logical operator and two bits, each separated by a single space.

Output
For each query, output the result. Your last line of output must contain a newline character.

Sample Input 1
```
1
2
AND 1 0
OR 0 1
```

Sample Output 1
```
0
1
```

Sample Input 2
```
2
AND 1 1
OR 1 0
AND 1 0
0
```

Sample Output 2
```
1
1
0
```

Sample Input 3
```
3
AND 1 1
OR 1 0
```

Sample Output 3
```
1
1
```

Explanation

In **Sample Input 1**, the input type is 1 and **N** equals to 2. So there are 2 queries that we need to process:

- The first query is "`AND 1 0`". The binary operator here is "AND" and the two bits are `1` and `0`. The result of this operation is 0. (`1 AND 0 = 0`).

- 2nd operation is "`OR 0 1`". The binary operation here is "OR" and the two bits are `0` and `1`. The result of this operation is 1 (`0 OR 1 = 1`).

We will omit the logical operation explanation until later.

In **Sample Input 2**, the input type is 2. It means that we need to read until a special character '0'. We have 3 queries to be processed in Sample Input 2 before the special character appears (a single line consisting of only the number "0").

In **Sample Input 3**, the input type is 3. It means that we need to read until the end of file. We have 2 queries to be processed in Sample Input 3.

For all sample inputs, we need to take note of how the logical operators "AND" and "OR" works. The bits given are either "1" or "0", which we can read as "TRUE" or "FALSE" or something similar. The operators and results correspond to the following truth tables:

Truth Table for "AND"

First Bit	Second Bit	Result
0	0	0
0	1	0
1	0	0
1	1	1

Truth Table for "OR"

First Bit	Second Bit	Result
0	0	0
0	1	1
1	0	1
1	1	1