

Count the Operations

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

In a class, Ania wrote a program which does some work for all integers from 0 to $n - 1$:

```
for (int i = 0; i < n; i++) {  
    work (i);  
}
```

The teacher Petia looked at the program and said that it would work for too long, as the maximum value of n is 10^9 .

In response, Ania decided to do the work only for a portion of the numbers. Her new program looks as follows:

```
for (int i = 0; i < n; i++) {  
    if (i == x_1) i = y_1;  
    if (i == x_2) i = y_2;  
    ...  
    if (i == x_k) i = y_k;  
    work (i);  
}
```

How many operations does this program perform for the given n , and does it terminate at all? The operations we count are assignments ($i = 0$; $i++$; $i = y_1$; ...), comparisons ($i < n$; $i == x_1$; ...), and the work itself (`work (i)`).

Input

The input contains the program written by Ania. It is formatted exactly as shown in the examples: in particular, the opening curly bracket is on the line with `for`, there is a space in front of every opening round bracket, and the indentation is four spaces. The input does not contain spaces at line ends.

The number of lines with conditionals: $0 \leq k \leq 10^5$. The constraints on the values: $0 \leq x_i, y_i < n \leq 10^9$, other than that, the values can be arbitrary. The line with `work` follows once after all the lines with `if`.

Output

Print the number of operations that the given program performs. If the program never terminates, print -1.

Examples

<i>standard input</i>	<i>standard output</i>	<i>operations</i>
<pre>for (int i = 0; i < 3; i++) { work (i); }</pre>	11	<pre>i = 0; 0 < 3; work (0); i++; 1 < 3; work (1); i++; 2 < 3; work (2); i++; 3 < 3.</pre>
<pre>for (int i = 0; i < 5; i++) { if (i == 3) i = 1; if (i == 1) i = 3; if (i == 3) i = 4; work (i); }</pre>	16	<pre>i = 0; 0 < 5; 0 == 3; 0 == 1; 0 == 3; work (0); i++; 1 < 5; 1 == 3; 1 == 1; i = 3; 3 == 3; i = 4; work (4); i++; 5 < 5.</pre>