

Число операций

Имя входного файла: *стандартный ввод*
Имя выходного файла: *стандартный вывод*
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

На занятии кружка Аня написала программу, которая для всех чисел от 0 до $n - 1$ проделывает работу:

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

Преподаватель Петя посмотрел на программу и сказал, что она будет работать слишком долго, ведь максимальное значение n равно 10^9 .

Тогда Аня решила перебирать не все числа, а только некоторые. Её новая программа выглядит так:

```
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);  
}
```

Сколько операций теперь делает эта программа для заданного числа n , и завершает ли она вообще работу когда-нибудь? Операциями следует считать присваивания ($i = 0$; $i++$; $i = y_1$; ...), сравнения ($i < n$; $i == x_1$; ...) и работу (`work (i)`).

Формат входных данных

Во вводе задана программа, которую написала Аня. Программа отформатирована ровно так, как показано в примерах: в частности, открывающая фигурная скобка находится на строке с `for`, перед открывающей круглой скобкой всегда стоит пробел, а отступы состоят из четырёх пробелов. Во вводе нет пробелов в конце строк.

Число строк с условиями: $0 \leq k \leq 10^5$. Ограничения на числа: $0 \leq x_i, y_i < n \leq 10^9$, в остальном числа могут быть любыми. Строка с `work` следует один раз после всех строк с `if`.

Формат выходных данных

Выведите, сколько операций делает заданная программа. Если программа никогда не завершит работу, выведите -1.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>	<i>операции</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>