Problem F
Halting Problem

Time Limit: 3 seconds

A unique law is enforced in the Republic of Finite Loop. Under the law, programs that never halt are regarded as viruses. Releasing such a program is a cybercrime. So, you want to make sure that your software products always halt under their normal use.

It is widely known that there exists no algorithm that can determine whether an arbitrary given program halts or not for a given arbitrary input. Fortunately, your products are based on a simple computation model given below. So, you can write a program that can tell whether a given program based on the model will eventually halt for a given input.

The computation model for the products has only one variable x and $N + 1$ states, numbered 1 through $N + 1$. The variable x can store any integer value. The state $N + 1$ means that the program has terminated. For each integer i ($1 \leq i \leq N$), the behavior of the program in the state i is described by five integers a_i, b_i, c_i, d_i and e_i (c_i and e_i are indices of states).

On start of a program, its state is initialized to 1, and the value of x is initialized by x_0, the input to the program. When the program is in the state i ($1 \leq i \leq N$), either of the following takes place in one execution step:

- if x is equal to a_i, the value of x changes to $x + b_i$ and the program state becomes c_i;
- otherwise, the value of x changes to $x + d_i$ and the program state becomes e_i.

The program terminates when the program state becomes $N + 1$.

Your task is to write a program to determine whether a given program eventually halts or not for a given input, and, if it halts, to compute how many steps are executed. The initialization is not counted as a step.

Input

The input consists of a single test case of the following format.

$N\ x_0$
$a_1\ b_1\ c_1\ d_1\ e_1$
$\vdots$
$a_N\ b_N\ c_N\ d_N\ e_N$

The first line contains two integers N ($1 \leq N \leq 10^5$) and x_0 ($-10^{13} \leq x_0 \leq 10^{13}$). The number of the states of the program is $N+1$. x_0 is the initial value of the variable x. Each of the next N lines contains five integers a_i, b_i, c_i, d_i and e_i that determine the behavior of the program when it is in the state i. a_i, b_i and d_i are integers between -10^{13} and 10^{13}, inclusive. c_i and e_i are integers between 1 and $N+1$, inclusive.

Output

If the given program eventually halts with the given input, output a single integer in a line which is the number of steps executed until the program terminates. Since the number may be very large, output the number modulo $10^9 + 7$.

Output -1 if the program will never halt.

Sample Input 1

```
2 0
5 1 2 1 1
10 1 3 2 2
```

Sample Output 1

```
9
```

Sample Input 2

```
3 1
0 1 4 2 3
1 0 1 1 3
3 -2 2 1 4
```

Sample Output 2

```
-1
```

Sample Input 3

```
3 3
1 -1 2 2 2
1 1 1 -1 3
1 1 4 -2 1
```

Sample Output 3

```
-1
```

Sample Input 4	Sample Output 4
13 3 15 -10 4 4 2 19 0 4 4 3 23 -20 4 4 1 6 1 2 1 5 6 17 3 -1 6 3 2 9 4 7 7 0 9 4 8 11 -5 9 4 6 6 1 7 1 10 6 5 8 -8 11 1 2 11 1 12 6 1 14 2 13 6 -4 11 1 12	26