

Problem A. Tree Puzzle

Input file: puzzle.in
Output file: puzzle.out
Time limit: 2 seconds
Memory limit: 256 megabytes

You have found a strange puzzle that looks like a full binary tree of height N (there are N nodes on the path from the root to any leaf). Nodes of the tree are numbered in the following way: the root node has number 1, any node x which is not a leaf has two children with numbers $2x$ (let's call it left child) and $2x + 1$ (right child).

Each node has its own state which can be 0 or 1. You can alter the states by starting a *flipping reaction* from any node. The flipping reaction then proceeds as follows. Firstly, the state of the node being flipped is changed to the opposite. After that, if this node is not a leaf, a chain reaction continues with one of its children: if the state of the node was 0 before the flip, the flipping reaction continues with its left child, and in the other case, with its right child. Thus the flipping reaction proceeds along a path from the starting node to exactly one leaf.

After starting a flipping reaction from some node, you should wait until the whole reaction is finished before starting the next one. Each node can be the starting node of a flipping reaction any number of times.

You are given the initial states of all nodes. Your goal is to set all states to 0 by starting the minimal possible number of flipping reactions.

As the number of nodes can be very large, we can not directly give the states in the input file. Luckily, you can get all states using the following algorithm: in the input file you are given integers a, b, c, p and T , where p is *prime* and b is *positive*. Using these numbers, you can generate the sequence x_i :

$$\begin{aligned}x_1 &= a \\x_i &= (x_{i-1} \cdot b + c) \bmod p \quad \text{where } i > 1\end{aligned}$$

For each node i , if $x_i \geq T$, the initial state of i -th node is 1, otherwise it's 0.

Input

The only line of input contains six integers N, a, b, c, p and T ($1 \leq N \leq 60, 0 \leq a < p, 1 \leq b < p, 0 \leq c < p, 2 \leq p \leq 1\,000\,000, 0 < T < p$). It is guaranteed that p is prime.

Output

Print the minimal possible number of flipping reactions needed to set the state of all nodes to 0.

Examples

puzzle.in	puzzle.out
3 3 2 1 5 2	4
4 4 1 0 7 3	8