Problem E. Cryptographic Argument

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

To enter the spaceship security system, the astromechanical droid R2-D2 uses a secret cryptographic algorithm, one step of which solves the following problem:

Consider a thin paper strip with $n = 2^k$ cells located in one row. The cells are numbered 0 through $n - 1$. Fold the strip in half, the right half under the left one, so that the cell numbered $2^{k-1} - j$ will be above the cell numbered $2^{k-1} + j - 1$ for all j from 1 to 2^{k-1}. Then fold the resulting strip, which has how length 2^{k-1}, again and again in the same way until the length of the strip is one. In the resulting folded strip, the cells are placed above each other in some order. Let the sequence $a_0, a_1, \ldots, a_{n-1}$ be the numbers of the cells from top to bottom.

When the droid connects to the spaceship system, it is given an integer n, and after that, it is asked several queries to get verified. Each query is a segment $[l, r]$ $(0 \le l \le r < n)$ for which the droid must compute the following expression, in which operations + and $\oplus$ alternate:

$$F(l, r) = a_l + a_{l+1} \oplus a_{l+2} + a_{l+3} \oplus a_{l+4} + a_{l+5} \oplus \ldots a_r.$$

Additional care must be taken since operation + has higher priority than $\oplus$ if l is even, and operation $\oplus$ has higher priority than + otherwise. If the droid doesn't answer correctly these queries in one second, he is revealed.

R2-D2 argued with C-3PO that the latter couldn't solve this problem. C-3PO is a protocol droid, it knows six million forms of communication, but nothing about cryptography. Help him to solve the problem in order to surprise R2-D2.

In order to emulate a security system, R2-D2 suggested an algorithm for choosing the segments $[l, r]$. He also suggested to check only the results of hashing $F(l, r)$, but not the values $F(l, r)$ themselves. Let the queries be numbered 0 through $m - 1$. Then these values are computed as follows:

- $h_{j+1} = ((l_j \oplus r_j \oplus h_j \oplus F(l_j, r_j)) + c) \bmod 1\,000\,000\,007$;
- $l_{j+1} = ((l_j \oplus a \oplus h_{j+1}) \bmod (n + 1)) \bmod n$;
- $r_{j+1} = ((r_j \oplus b \oplus h_{j+1}) \bmod (n - l_{j+1})) + l_{j+1}$.

Here, h is the value of the hash function, and $h_0 = 0$. Your task is to compute the final hash value h_m.

Input

The first line of input contains an integer k $(0 \le k \le 30)$.

The second line contains three integers m, l_0, r_0: the number of queries and the bounds of the initial query $(1 \le m \le 10^7,\ 0 \le l_0 \le r_0 < n)$.

The third line contains three integers a, b and c: the parameters of the generating algorithm $(0 \le a, b, c < 2^{30})$.

Output

Print one integer h_m: the value of the hash function after processing all queries.

Examples

standard input	standard output
3 1 2 6 3 4 5	7
3 709193 4 5 273035200 65685838 991992535	156951996

Note

Operation $\oplus$ means exclusive or.

The results of operations with higher priority must be computed earlier. Operations with the same priority must be processed in the order from left to right.

A strip of length $2^3 = 8$ folds like in the picture.