

Problem G. Random Shuffle Ranking

Input file: *standard input*
Output file: *standard output*
Time limit: 5 seconds (*6 seconds for Java*)
Memory limit: 256 mebibytes

Yakov works in Yandex ranking team in the core of the web search itself.

On the eve of an important release, Yakov has noticed that one of the reranking rules is faulty: it rearranges documents in a wrong way. As a conscientious employee, Yakov can not let such a failure affect users' search experience.

But the new binaries are already deployed to production servers, and there is little time left before the faulty functionality will be turned on. So, Yakov decided to resort to desperate measures: he will try to fix everything by changing parameters in server configuration files.

Yakov noticed that after applying the new reranking rules, the documents are shuffled using the `Shuffle` function given below. The default value for `salt` in the configuration file is zero, but Yakov is about to change it.

```
void Shuffle(int a[N], int salt) {
    for (int i = 0; i < N; ++i) {
        int j = i ^ salt; // bitwise xor with salt
        if (i < j) {
            swap(a[i], a[j]);
        }
    }
}
```

You can help Yakov by determining how useful this function is.

Consider a sequence of N integers a_i . Let us denote an array produced by `Shuffle(a, x)` as b_x .

Consider a sequence of arrays b_x for $x = 0, \dots, N - 1$. Your task is to find the sorting permutation p_i ($0 \leq p_i < N$) such that the sequence $b_{p_0}, b_{p_1}, \dots, b_{p_{N-1}}$ is ordered so that b_{p_i} is lexicographically less or equal to $b_{p_{i+1}}$ for $i = 0, \dots, N - 2$. And there is an additional constraint on the sequence p : if for some p_i and p_j , the arrays b_{p_i} and b_{p_j} are equal to each other, then the inequality $p_i < p_j$ holds.

As N in consideration could be large, you are to output the polynomial hash of the sought permutation: $(p_0 \cdot q^{N-1} + p_1 \cdot q^{N-2} + \dots + p_{N-2} \cdot q + p_{N-1}) \bmod 2^{32}$, where $q = 10^9 + 7$.

Input

The input consists of several test cases. Each test case comprises nine space-separated integers on a single line: $N, a_{-3}, a_{-2}, a_{-1}, A, B, C, D$ and M .

The sequence a_i ($0 \leq i < N$) in consideration is generated in the following manner:

$$a_i = (A \cdot a_{i-3} + B \cdot a_{i-2} + C \cdot a_{i-1} + D) \bmod M.$$

The constraints are:

- $N = 2^k, 0 \leq k \leq 17$,
- $0 \leq a_{-3}, a_{-2}, a_{-1}, A, B, C, D \leq 10^9$,
- $1 \leq M \leq 10^9$,
- the input consists of no more than 2^{17} test cases,
- the sum of all N in a single input does not exceed 2^{21} .

Output

For each test case, output a single integer: the polynomial hash of the sought permutation.

Example

standard input	standard output
4 0 0 1 2 0 3 1 4	1628479554
4 0 0 1 0 0 1 1 4	2008884034