I – Palindromes

Memory limit: 1024 MB
Time limit: 3 s

EUC 2026
8.02.2026

In this problem we consider strings over the binary alphabet $\{\texttt{a}, \texttt{b}\}$. A string is called a *palindrome* if it reads the same from left to right and from right to left. For example, the strings `a`, `aba`, and `babbab` are palindromes, but `abab` is not.

There is an unknown binary string s of length N over the alphabet $\{\texttt{a}, \texttt{b}\}$. We only know that its prefixes (that is, initial fragments) of lengths $a_1, \ldots, a_n$ are palindromes. In how many ways can the string s be recovered? The answer can be quite large; it suffices to return its remainder modulo m.

Note that the sought string s may have other prefixes being palindromes, in addition to the ones of lengths $a_1, \ldots, a_n$.

Input

The first line contains three integers N, n, and m ($1 \leq N \leq 10^{18}$, $1 \leq n \leq 500\,000$, $2 \leq m \leq 10^9$) that represent the length of the string, the number of palindromic prefixes of the string, and the number used to compute the result.

The second line contains a strictly increasing sequence of n integers $a_1, a_2, \ldots, a_n$ (with $1 \leq a_i \leq N$) that denote the lengths of the palindromic prefixes of the sought string.

Output

The output should contain one non-negative integer, the number of strings of length N that have a palindromic prefix of each length $a_1, \ldots, a_n$, modulo m.

Example

For the input data:

```
10 3 100
2 5 9
```

the correct result is:

```
8
```

Explanation: There are 8 strings of length 10 whose prefixes of lengths 2, 5, and 9 are palindromes:

1. `aaaaaaaaaa`
2. `aaaaaaaaab`
3. `aabaaabaaa`
4. `aabaaabaab`
5. `bbabbbabba`
6. `bbabbbabbb`
7. `bbbbbbbbba`
8. `bbbbbbbbbb`