B
Periodic points

Computing the number of fixed points and, more generally, the number of periodic orbits within a dynamical system is a question attracting interest from different fields of research. However, dynamics may turn out to be very complicated to describe, even in seemingly simple models. In this task you will be asked to compute the number of periodic points of period n of a piecewise linear map f mapping the real interval $[0, m]$ into itself. That is to say, given a map $f : [0, m] \to [0, m]$ you have to calculate the number of solutions to the equation $f^n(x) = x$ for $x \in [0, m]$, where f^n is the result of iterating function f a total of n times, i.e.

$$f^n = \overbrace{f \circ .. \circ f \circ f}^{n\ f\text{'s}},$$

where $\circ$ stands for the composition of maps, $(g \circ h)(x) = g(h(x))$.

Fortunately, the maps you will have to work with satisfy some particular properties:

- m will be a positive integer and the image of every integer in $[0, m]$ under f is again an integer in $[0, m]$, that is, for every $k \in \{0, 1, \ldots, m\}$ we have that $f(k) \in \{0, 1, \ldots, m\}$.
- For every $k \in \{0, 1, \ldots, m-1\}$, the map f is linear in the interval $[k, k+1]$. This means that for every $x \in [k, k+1]$, its image satisfies $f(x) = (k + 1 - x)f(k) + (x - k)f(k + 1)$, which is equivalent to its graph on $[k, k + 1]$ being a straight line segment.

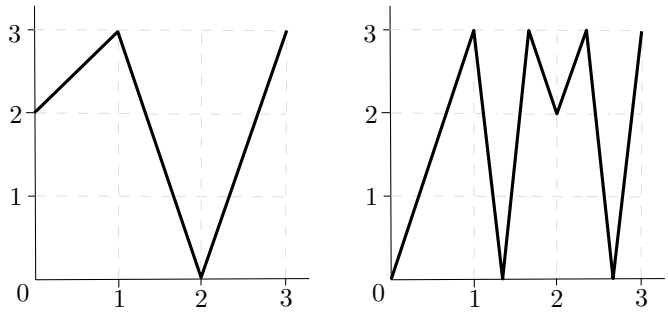


Figure 1: Graphs of the third map in the sample input, f_3 (left), and of its square, f_3^2 (right).

Since there might be many periodic points you will have to output the result modulo an integer.

Input

The input consists of several test cases, separated by single blank lines. Each test case begins with a line containing the integer m ($1 \leq m \leq 80$). The following line describes the map f; it contains the $m+1$ integers $f(0), f(1), \ldots, f(m)$, each of them between 0 and m inclusive. The test case ends with a line containing two integers separated by a blank space, n ($1 \leq n \leq 5\,000$) and the modulus used to compute the result, *mod* ($2 \leq mod \leq 10\,000$).

The input will finish with a line containing `0`.

Output

For each case, your program should output the number of solutions to the equation $f^n(x) = x$ in the interval $[0, m]$ modulo *mod*. If there are infinitely many solutions, print `Infinity` instead.

Sample Input

```
2
2 0 2
2 10

3
0 1 3 2
1 137

3
2 3 0 3
20 10000

0
```

Sample Output

```
4
Infinity
9074
```