Hashing

Time Limit: 5 Second
Memory Limit: 256 MB

As a world class computer scientist, Mattox is particularly interested in hashing tables. Today, he proposed a new hashing algorithm described as follows:

1. The hash table contains n entries $0, \ldots, n-1$, and each entry is a linked list of size no greater than m. As a result, the total number of elements that can be stored in the hash table is $n \times m$.
2. For each new value x, the insertion process is as follows:
   (a) Calculate its hash value $h(x) \in \{0, \ldots, n-1\}$.
   (b) Set current entry to be $h(x)$.
   (c) If current entry has less than m items, append x to current entry. Otherwise go to next entry (i.e. $(i+1) \mod n$-th entry) and repeat the process until the condition is met.
   (d) Insert a new entry into the hash table if all entries are full.

As a talented student in his lab, Mattox wants you to implement the most important step in this algorithm: given a hash table H, a hash function h, and a new value x, find the entry that x should be inserted into according to the algorithm. To preserve the properties of a hash table, the target runtime of the step should be $O(1)$.

Input

The first line of input contains three integers n, m, and k ($1 \leq n, m \leq 10^6$, $1 \leq k \leq \min(nm, 2 \times 10^6)$) - the number of entries in the hash table, the maximum length of each entry, and the number of insertions. The hash table is initially empty.

The second line contains two integers a and b ($1 \leq a, b \leq 10^9$). The hash function is defined as $h(x) = (ax+b) \mod n$.

The next k lines describe the values to be inserted. Each line contains a single integer v ($1 \leq v \leq 10^9$), denoting the value to be inserted.

Output

For each insertion, output a single integer denoting the entry to insert the new value.

Sample Inputs

```
5 3 8
1 1
1
1
1
2
3
4
5
6
```

Sample Outputs

```
2
2
2
3
4
0
1
3
```