code jam
System.out.println("hello, world!");

Practice Mode

Contest scoreboard | Sign in

Round 1C 2008

A. Text Messaging Outrage

B. Ugly Numbers

C. Increasing Speed Limits

Contest Analysis

Questions asked 4

Submissions

Text Messaging Outrage

5pt	Not attempted **2204/2255 users** correct (98%)
10pt	Not attempted **1402/2194 users** correct (64%)

Ugly Numbers

10pt	Not attempted **554/1040 users** correct (53%)
25pt	Not attempted **82/318 users** correct (26%)

Increasing Speed Limits

15pt	Not attempted **398/716 users** correct (56%)
35pt	Not attempted **49/312 users** correct (16%)

Top Scores

austrin	100
Baltazar	100
vepifanov	100
elizarov	100
xhl.kogitsune	100
ivan.popelyshev	100
SergeyRogulenko	100
Vasyl	100
slex	100
frankyym	100

Problem C. **Increasing Speed Limits**

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 15 points	Solve C-small
Large input 35 points	Solve C-large

Problem

You were driving along a highway when you got caught by the road police for speeding. It turns out that they've been following you, and they were amazed by the fact that you were accelerating the whole time without using the brakes! And now you desperately need an excuse to explain that.

You've decided that it would be reasonable to say "all the speed limit signs I saw were in increasing order, that's why I've been accelerating". The police officer laughs in reply, and tells you all the signs that are placed along the segment of highway you drove, and says that's unlikely that you were so lucky just to see some part of these signs that were in increasing order.

Now you need to estimate that likelihood, or, in other words, find out how many different subsequences of the given sequence are strictly increasing. The empty subsequence does not count since that would imply you didn't look at any speed limits signs at all!

For example, (1, 2, 5) is an increasing subsequence of (1, 4, 2, 3, 5, 5), and we count it twice because there are two ways to select (1, 2, 5) from the list.

Input

The first line of input gives the number of cases, **N**. **N** test cases follow. The first line of each case contains **n**, **m**, **X**, **Y** and **Z** each separated by a space. **n** will be the length of the sequence of speed limits. **m** will be the length of the generating array A. The next **m** lines will contain the **m** elements of A, one integer per line (from A[0] to A[**m**-1]).

Using A, **X**, **Y** and **Z**, the following pseudocode will *print* the speed limit sequence in order. mod indicates the remainder operation.

```
for i = 0 to n-1
  print A[i mod m]
  A[i mod m] = (X * A[i mod m] + Y * (i + 1)) mod Z
```

Note: The way that the input is generated has nothing to do with the intended solution and exists solely to keep the size of the input files low.

Output

For each test case you should output one line containing "Case #**T**: **S**" (quotes for clarity) where **T** is the number of the test case and **S** is the number of non-empty increasing subsequences mod 1 000 000 007.

Limits

$1 \leq \mathbf{N} \leq 20$
$1 \leq \mathbf{m} \leq 100$
$0 \leq \mathbf{X} \leq 10^9$
$0 \leq \mathbf{Y} \leq 10^9$
$1 \leq \mathbf{Z} \leq 10^9$
$0 \leq \mathbf{A[i]} < \mathbf{Z}$

Small dataset

$1 \leq \mathbf{m} \leq \mathbf{n} \leq 1000$

Large dataset

$1 \leq \mathbf{m} \leq \mathbf{n} \leq 500\ 000$

Sample

```
Input                   Output

2                       Case #1: 15
5 5 0 0 5               Case #2: 13

1
2
```

```
1
2
3
6 2 2 1000000000 6
1
2
```

The sequence of speed limit signs for case 2 should be 1, 2, 0, 0, 0, 4.