Definite Random Walks

by zemen

Alex has a board game consisting of:

- A *chip* for marking his current location on the board.
- n *fields* numbered from 1 to n. Each position i has a value, f_i, denoting the *next* position for the chip to jump to from that field.
- A *die* with m faces numbered from 0 to $m-1$. Each face j has a probability, p_j, of being rolled.

Alex then performs the following actions:

- Begins the game by placing the chip at a position in a field randomly and with equiprobability.
- Takes k turns; during each turn he:
  - Rolls the die. We'll denote the number rolled during a turn as d.
  - Jumps the chip d times. Recall that each field contains a value denoting the *next* field number to jump to.
- After completing k turns, the game ends and he must calculate the respective probabilities for each field as to whether the game ended with the chip in that field.

Given n, m, k, the game board, and the probabilities for each *die* face, print n lines where each line i contains the probability that the chip is on field i at the end of the game.

Note: All the probabilities in this task are rational numbers modulo $M = 998244353$. That is, if the probability can be expressed as the irreducible fraction $\frac{p}{q}$ where $q \bmod M \neq 0$, then it corresponds to the number $(p \times q^{-1}) \bmod M$ (or, alternatively, $p \times q^{-1} \equiv x(\bmod M)$). Click here to learn about *Modular Multiplicative Inverse*.

Input Format

The first line contains three space-separated integers describing the respective values of n (the number of positions), m (the number of die faces), and k (the number of turns).
The second line contains n space-separated integers describing the respective values of each f_i (i.e., the index of the field that field i can transition to).
The third line contains m space-separated integers describing the respective values of each p_j (where $0 \leq p_j < M$) describing the probabilities of the faces of the m-sided die.

Constraints

- $1 \leq n \leq 6 \times 10^4$
- $4 \leq m \leq 10^5$
- $1 \leq k \leq 1000$
- $1 \leq i, f_i \leq n$
- $0 \leq p_j < M$
- The sum of $p_j \bmod M$ is 1

Note: The time limit for this challenge is doubled for *all* languages. Read more about standard time limits at our environment page.

Output Format

Print n lines of output in which each line i contains a single integer, x_i (where $0 \leq x_i < M$), denoting the probability that the chip will be on field i after k turns.

Sample Input 0

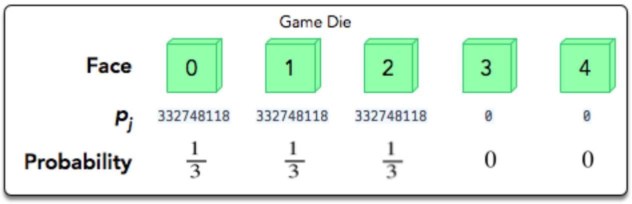
```
4 5 1
2 3 2 4
332748118 332748118 332748118 0 0
```

Sample Output 0

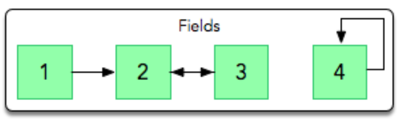
```
582309206
332748118
332748118
748683265
```

Explanation 0

The diagram below depicts the respective probabilities of each *die* face being rolled:



The diagram below depicts each field with an arrow pointing to the *next* field:



There are four equiprobable initial fields, so each field has a $\frac{1}{4}$ probability of being the chip's initial location. Next, we calculate the probability that the chip will end up in each field after $k = 1$ turn:

1. The only way the chip ends up in this field is if it never jumps from the field, which only happens if Alex rolls a 0. So, this field's probability is $\frac{1}{4} \cdot \frac{1}{3} = \frac{1}{12}$. We then calculate and print the result of $\frac{1}{12} \bmod 998244353 = 582309206$ on a new line.
2. The chip can end up in field 2 after one turn in the following scenarios:
   - Start in field 1 and roll a 1, the probability for which is $\frac{1}{4} \cdot \frac{1}{3} = \frac{1}{12}$.
   - Start in field 2 and roll a 0 or a 2, the probability for which is $\frac{1}{4} \cdot \frac{2}{3} = \frac{2}{12}$.
   - Start in field 3 and roll a 1, the probability for which is $\frac{1}{4} \cdot \frac{1}{3} = \frac{1}{12}$.

   After summing these probabilities, we get a total probability of $\frac{1}{12} + \frac{2}{12} + \frac{1}{12} = \frac{1}{3}$ for the field. We then calculate and print the result of $\frac{1}{3} \bmod 998244353 = 332748118$ on a new line.
3. The chip can end up in field 3 after one turn in the following scenarios:
   - Start in field 1 and roll a 2, the probability for which is $\frac{1}{4} \cdot \frac{1}{3} = \frac{1}{12}$.
   - Start in field 2 and roll a 1, the probability for which is $\frac{1}{4} \cdot \frac{1}{3} = \frac{1}{12}$.
   - Start in field 3 and roll a 0 or a 2, the probability for which is $\frac{1}{4} \cdot \frac{2}{3} = \frac{2}{12}$.

   After summing these probabilities, we get a total probability of $\frac{1}{12} + \frac{1}{12} + \frac{2}{12} = \frac{1}{3}$ for the field. We then calculate and print the result of $\frac{1}{3} \bmod 998244353 = 332748118$ on a new line.

4. If the chip is initially placed in field 4, it will always end up in field 4 regardless of how many turns are taken (because this field loops back onto itself). Thus, this field's probability is $\frac{1}{4}$. We then calculate and print the result of $\frac{1}{4} \bmod 998244353 = 748683265$ on a new line.

Submissions:39

Max Score:100
Difficulty: Expert

Rate This Challenge:

More

Current Buffer (saved locally, editable)

Java 7

```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int n = in.nextInt();
        int m = in.nextInt();
        int k = in.nextInt();
        int[] f = new int[n];
        for(int f_i=0; f_i < n; f_i++){
            f[f_i] = in.nextInt();
        }
        int[] p = new int[m];
        for(int p_i=0; p_i < m; p_i++){
            p[p_i] = in.nextInt();
        }
        // your code goes here
    }
}

```

Line: 1 Col: 1

Upload Code as File   Test against custom input

Run Code   Submit Code