2024 Canadian Computing Olympiad
Day 2, Problem 1
Infiltration

Time Limit: 1 second

Problem Description

Ondrej and Edward are spies and they are going to take down the evil organization AQT. To do so, they will need to infiltrate into the AQT base. The base can be modelled as a tree with $N = 100$ rooms, labelled from 0 to $N - 1$. Ondrej and Edward's plan to infiltrate the base is to first get kidnapped and then meet up together before executing their plan. When kidnapped, the two will be placed into different rooms unknown to each other. Once they are placed into the rooms, they will both break free at midnight and try to meet up with each other before executing their plan.

Their plan to meet up is as follows. At every odd minute, Ondrej can choose to stay at his current room or move to an adjacent room. At every even minute, Edward can choose to stay at his current room or move to an adjacent room.

A strategy is defined as the following. Let $V(A, R, T)$ denote the room agent A should be at assuming that they were at room R at midnight and it is currently T minutes after midnight. The strategy should match the conditions above. The agents are said to meet up at time $t(o, e)$, which is the first time where $V(\text{Ondrej}, o, t(o, e)) = V(\text{Edward}, e, t(o, e))$.

Ondrej and Edward want to meet up as fast as possible, relative to the distance between their two starting rooms. The distance $d(o, e)$ is the minimum number of corridors that must be traversed to reach o from e. Please help find a strategy that minimizes the maximum $\frac{t(o,e)}{d(o,e)}$ across all pairs of different rooms o and e.

Input Specification

The first line of input will contain N ($N = 100$). [Post-CCO edit: If the value of N is anything other than 100, exit the program immediately.]

The next $N - 1$ lines will each contain two space-separated integers, denoting the labels of two rooms with a bidirectional corridor between them.

Output Specification

First output a positive number T, the number of entries per starting room. Note that $T \leq 1440$ must be satisfied, otherwise you will be awarded no points.

Then, output Ondrej's strategy, followed by Edward's strategy.

To output an agent's strategy, output N lines, where the n-th line (starting from 0) represents the agent's path if they start at room n. For each line, output T spaced integers: The room label that the agent should be in at time $1, 2, \ldots, T$.

Scoring

If the output is invalid or there exists a test case and a pair of different rooms o and e where the agents do not meet at or before time T, then no points will be awarded.

Otherwise, let S be the maximum among all test cases and pairs of o and e ($o \neq e$) of the value of $\frac{t(o,e)}{d(o,e)}$. The following table shows how the available 25 marks are distributed:

Score	Bounds on S
3	$200 < S \leq 1440$
6	$100 < S \leq 200$
8	$50 < S \leq 100$
10	$40 < S \leq 50$
12	$30 < S \leq 40$
15	$25 < S \leq 30$
17	$20 < S \leq 25$
18	$19 < S \leq 20$
19	$18 < S \leq 19$
20	$17 < S \leq 18$
21	$16 < S \leq 17$
22	$15 < S \leq 16$
25	$S \leq 15$

Sample Input 1

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

Output for Sample Input 1

```
8
2 2 4 4 1 1 1 1
1 1 1 1 1 1 1 1
3 3 2 2 3 3 2 2
3 3 2 2 0 0 2 2
4 4 4 4 2 2 2 2
0 2 2 3 3 3 3 2
```

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

Explanation of Output for Sample Input 1

Note that this is an invalid test case as $N \neq 100$, so it will not appear in the test cases when judging. The output for the test case is valid. Note that this would not score any points because if Ondrej starts at room 1 and Edward starts at room 3, then they will never meet each other.

2024 Canadian Computing Olympiad
Day 2, Problem 2
Heavy Light Decomposition

Time Limit: 4 seconds

Problem Description

In an array containing only positive integers, we say an integer is heavy if it appears more than once in the array, and light otherwise.

An array is good if the integers in the array alternate between light and heavy.

Given an array $a_1, \ldots, a_N$, count the number of ways to partition it into some number of contiguous subarrays such that each subarray, when considered as an array on its own, is good. As the answer may be large, output it modulo 1 000 003.

Input Specification

The first line of input contains a single integer, N.

The next line contains N integers $a_1, \ldots, a_N (1 \leq a_i \leq N)$.

Marks Awarded	Bounds on N	Additional Constraints
3 marks	$2 \leq N \leq 50\,000$	For each i, $a_i \leq 26$.
4 marks	$2 \leq N \leq 5\,000$	No additional constraints.
5 marks	$2 \leq N \leq 500\,000$	If i is odd, then $a_i = 1$.
6 marks	$2 \leq N \leq 500\,000$	Any number appears at most twice in the array.
7 marks	$2 \leq N \leq 500\,000$	No additional constraints.

Output Specification

The number of ways to partition the array into good contiguous subarrays, modulo 1 000 003.

Sample Input 1

```
5
1 2 3 2 3
```

Output for Sample Input 1

```
4
```

Explanation of Output for Sample Input 1

There are four valid partitions of [1, 2, 3, 2, 3]:

- [1], [2], [3], [2], [3]
- [1], [2, 3, 2], [3]
- [1], [2], [3, 2, 3]
- [1, 2, 3, 2], [3]

Sample Input 2

```
5
1 2 1 3 1
```

Output for Sample Input 2

```
6
```

2024 Canadian Computing Olympiad
Day 2, Problem 3
Telephone Plans

Time Limit: 4 seconds

Problem Description

The "Dormi's Fone Service" is now the only telephone service provider in CCOland. There are N houses in CCOland, numbered from 1 to N. Each telephone line connects two distinct houses such that all the telephone lines that ever exist form a forest.

There is an issue where the phone lines are faulty, and each phone line only exists for a single interval of time. Two houses can call each other at a certain time if there is a path of phone lines that starts at one of the houses and ends in the other house at that time.

We would like to process Q queries of the following forms:

- `1 x y`: Add a phone line between houses x and y. It is guaranteed that a phone line between houses x and y was never added before.
- `2 x y`: Remove the phone line between houses x and y. It is guaranteed that a phone line currently exists between houses x and y.
- `3 t`: Compute the number of pairs of different houses that can call each other at some time between the current query and t queries ago. To be more clear, let G_q be the state of CCOland after the q-th query, where G_0 is the state of CCOland before any queries. If this is the s-th query, then count the number of pairs of houses that are connected in at least one of $G_{s-t}, G_{s-t+1}, \ldots, G_s$.

Also, some test cases may be encrypted. For the test cases that are encrypted, the arguments `x`, `y`, or `t` are given as the bitwise xor of the true argument and the answer to the last query of type `3` (if there have been no queries of type `3`, then the arguments are unchanged).

Input Specification

The first line of input will contain E ($E \in \{0, 1\}$). $E = 0$ denotes that the input is not encrypted, while $E = 1$ denotes that the input is encrypted.

The second line contains two space-separated integers N and Q, representing the number of houses in CCOland and the number of queries, respectively.

The next Q lines contain queries as specified above (queries are encrypted or not depending on E).

For the q-th query ($1 \leq q \leq N$), it is guaranteed that (after decrypting if $E = 1$) $1 \leq x, y \leq N$ for type `1` and `2` queries and $0 \leq t \leq q$ for type `3` queries.

Marks Awarded	Bounds on N	Bounds on Q	Encrypted?
3 marks	$1 \le N \le 30$	$1 \le Q \le 150$	$E = 0$
2 marks	$1 \le N \le 30$	$1 \le Q \le 150$	$E = 1$
4 marks	$1 \le N \le 2\,000$	$1 \le Q \le 6\,000$	$E = 0$
2 marks	$1 \le N \le 2\,000$	$1 \le Q \le 6\,000$	$E = 1$
4 marks	$1 \le N \le 100\,000$	$1 \le Q \le 300\,000$	$E = 0$
5 marks	$1 \le N \le 100\,000$	$1 \le Q \le 300\,000$	$E = 1$
5 marks	$1 \le N \le 500\,000$	$1 \le Q \le 1\,500\,000$	$E = 1$

Output Specification

For each query of type `3`, output the answer to the query on a new line.

Sample Input 1

```
0
4 12
3 0
1 1 2
3 0
1 1 3
3 0
3 5
2 2 1
3 0
3 8
1 1 4
3 0
3 11
```

Output for Sample Input 1

```
0
1
3
3
1
3
3
5
```

Explanation of Output for Sample Input 1

This test case is not encrypted.

For the 1st query, no pairs of different houses could have called each other.

For the 3rd query, only houses 1 and 2 could have called each other.

For the 5th query, $\{(1,2),(1,3),(2,3)\}$ is the set of pairs that could have called each other. The 6th query is the same.

For the 8th query, only houses 1 and 3 could have called each other.

For the 9th query, there is a point in time where $\{(1,2),(1,3),(2,3)\}$ could have called each other.

For the 11th query, the set of pairs that could have called each other is $\{(1,3),(1,4),(3,4)\}$.

For the 12th query, the set of pairs that could have called each other at any previous time is $\{(1,2),(1,3),(1,4),(2,3),(3,4)\}$.

Sample Input 2

```
1
4 12
3 0
1 1 2
3 0
1 0 2
3 1
3 6
2 1 2
3 3
3 9
1 2 7
3 3
3 8
```

Output for Sample Input 2

```
0
1
3
3
1
3
3
5
```

Explanation of Output for Sample Input 2

Encrypted version of sample 1.