

## 1 分组游戏

### 1.1 问题描述

有  $n$  个人，要分成若干组，对于第  $i$  个人，它所在的组的人数不能超过  $a_i$ ，问分组的方案数模 998244353 的值。

组与组之间是不可区分的，每组的人也是无序的，但人与人之间是可以区分的。

### 1.2 输入格式

第一行一个正整数  $n$ 。

接下来一行  $n$  个正整数  $a_i$ 。

#### 1.2.1 输出格式

一行一个整数表示答案。

### 1.3 样例

#### 1.3.1 样例输入一

```
3
2 3 3
```

#### 1.3.2 样例输出一

```
4
```

### 1.4 数据范围

对于 10% 的数据， $n \leq 10$ 。

对于 30% 的数据， $n \leq 50$ 。

对于 50% 的数据， $n \leq 300$ 。

对于 100% 的数据， $n \leq 2000, a_i \leq n$ 。

## Problem B. 排序 (sort.c/cpp/pas)

Input file:            `sort.in`  
Output file:          `sort.out`  
Time limit:          1 seconds  
Memory limit:        1 gigabytes

老虎和蒜头是好朋友。

老虎最近得到了一个排列  $a$ 。这一次，老虎决定将这个序列排序。

老虎的排序方法是这样的：每次他会选择排列  $a$  的一个子序列  $b$ ，删去这个子序列后，将这个子序列插回到原序列的最前面。

老虎想要考考蒜头，因此他想要蒜头回答出，至少需要几次操作可以将序列  $a$  排好序。同时，为了说明你不是随便报的，你需要输出操作的方案。

### Input

输入的第一行包括一个正整数  $n$ ，表示排列  $a$  的长度。

接下来一行  $n$  个数，其中的第  $i$  个数表示排列  $a$  中的第  $i$  个元素。

### Output

输出的第一行包括一个正整数  $m$ ，表示最少的操作次数。

接下来共包括  $m + 1$  行，其中第一行表示原序列，接下来的  $m$  行中的第  $i$  行表示第  $i$  次操作后的序列。

### Examples

<code>sort.in</code>	<code>sort.out</code>
4 4 1 2 3	1 4 1 2 3 1 2 3 4
4 2 1 4 3	2 2 1 4 3 1 4 2 3 1 2 3 4

在样例一中，我们选择子序列 1 2 3 即可。

在样例二中，两次选择的子序列分别是 1 4 和 1 2 3。

### Notes

对于 20% 的数据， $1 \leq n \leq 10$ ；

对于 50% 的数据， $1 \leq n \leq 20$ ；

对于另外 20% 的数据， $n$  在  $[1, 3000]$  中随机生成，排列在所有排列中随机生成；

对于 100% 的数据， $1 \leq n \leq 3000$ 。

如果你输出的第一行正确，可以得到 40% 的分数，但你仍然需要保证你的输出符合格式要求。

## Problem C. 网格 (grid.c/cpp/pas)

Input file:           grid.in  
Output file:          grid.out  
Time limit:          1 seconds  
Memory limit:        1 gigabytes

老虎和蒜头是好朋友。

老虎最近得到了一个  $n \times m$  的网格，网格上填着 0 至  $nm - 1$  间的非负整数，且每个数恰好出现一次。

现在老虎想要将这个网格还原成标准的形式。对于一个标准的  $n \times m$  的网格来说，其第  $i$  行第  $j$  列的元素恰好为  $(i - 1) \times m + (j - 1)$ 。

为了达成这个目标，老虎可以进行以下操作：

1. 对第  $i$  行向右进行  $k$  次 shift 操作，每次操作即所有元素向右移动一格，最后一个元素移动到第一格。
2. 对第  $i$  列向下进行  $k$  次 shift 操作，每次操作即所有元素向下移动一格，最后一个元素移动到第一格。

然而有许多初始填法超出了老虎还原的能力，因此他向蒜头寻求帮助。蒜头当然认为这很容易，但他太忙了，于是把任务交给你。

### Input

输入的第一行包括两个正整数  $n, m$ ，表示网格的大小。

接下来  $n$  行，每行  $m$  个数，表示网格初始的状况。

### Output

输出的第一行包括一个正整数  $N$ ，表示操作次数。你必须保证  $N \leq 10^5$ 。

接下来  $N$  行，每行输出一个操作。如果是向右进行操作，输出  $1 \ r \ k$  ( $1 \leq r \leq n, 0 \leq k \leq m$ )，表示将第  $r$  行向右 shift  $k$  次；否则输出  $2 \ c \ k$  ( $1 \leq c \leq m, 0 \leq k \leq n$ )，表示将第  $c$  列向下 shift  $k$  次。

如果不可能还原网格，你只需要输出  $-1$  即可。

Examples

grid.in	grid.out
2 4	2
4 2 3 0	2 1 1
1 5 6 7	1 1 1
4 2	7
2 3	1 1 1
5 0	2 1 1
4 1	1 2 1
6 7	1 3 1
	2 1 2
	1 1 1
	2 1 1

在样例一中，进行一次操作后的网格如下所示：

1 2 3 0  
4 5 6 7

Notes

- 对于 5% 的数据， $n, m \leq 3$ ;
- 对于 15% 的数据， $n, m \leq 5$ ;
- 对于 25% 的数据， $n, m \leq 8$ ;
- 对于 40% 的数据， $n, m \leq 10$ ;
- 对于 50% 的数据， $n, m \leq 20$ ;
- 对于另外 5% 的数据， $n = 2$ ;
- 对于另外 10% 的数据， $n = 3$ ;
- 对于另外 15% 的数据， $\min(n, m) \leq 5$ ;
- 对于 100% 的数据， $2 \leq n, m \leq 100$ 。