A. Toda 2

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

A group of n friends enjoys playing popular video game Toda 2. There is a rating system describing skill level of each player, initially the rating of the i-th friend is r_i.

The friends decided to take part in the championship as a team. But they should have equal ratings to be allowed to compose a single team consisting of all n friends. So the friends are faced with the problem: how to make all their ratings equal.

One way to change ratings is to willingly lose in some matches. Friends can form a party consisting of **two** to **five** (but not more than n) friends and play a match in the game. When the party loses, the rating of each of its members decreases by 1. A rating can't become negative, so $r_i = 0$ doesn't change after losing.

The friends can take part in multiple matches, each time making a party from any subset of friends (but remember about constraints on party size: from 2 to 5 members).

The friends want to make their ratings equal but as high as possible.

Help the friends develop a strategy of losing the matches so that all their ratings become equal and the resulting rating is maximum possible.

Input

The first line contains a single integer n ($2 \le n \le 100$) — the number of friends.

The second line contains n non-negative integers $r_1, r_2, ..., r_n$ ($0 \le r_i \le 100$), where r_i is the initial rating of the i-th friend.

Output

In the first line, print a single integer R — the final rating of each of the friends.

In the second line, print integer t — the number of matches the friends have to play. Each of the following t lines should contain n characters '`0`' or '`1`', where the j-th character of the i-th line is equal to:

- '`0`', if friend j should not play in match i,
- '`1`', if friend j should play in match i.

Each line should contain between two and five characters '`1`', inclusive.

The value t should not exceed 10^4, it is guaranteed that such solution exists.

Remember that you shouldn't minimize the value t, but you should maximize R. If there are multiple solutions, print any of them.

Examples

input
5 4 5 1 7 4

output
1 8 01010 00011

```
01010
10010
00011
11000
00011
11000
```

input

```
2
1 2
```

output

```
0
2
11
11
```

input

```
3
1 1 1
```

output

```
1
0
```