B. Tennis Game

Petya and Gena love playing table tennis. A single match is played according to the following rules: a match consists of multiple sets, each set consists of multiple serves. Each serve is won by one of the players, this player scores one point. As soon as one of the players scores t points, he wins the set; then the next set starts and scores of both players are being set to 0. As soon as one of the players wins the total of s sets, he wins the match and the match is over. Here s and t are some positive integer numbers.

To spice it up, Petya and Gena choose new numbers s and t before every match. Besides, for the sake of history they keep a record of each match: that is, for each serve they write down the winner. Serve winners are recorded in the chronological order. In a record the set is over as soon as one of the players scores t points and the match is over as soon as one of the players wins s sets.

Petya and Gena have found a record of an old match. Unfortunately, the sequence of serves in the record isn't divided into sets and numbers s and t for the given match are also lost. The players now wonder what values of s and t might be. Can you determine all the possible options?

Input

The first line contains a single integer n — the length of the sequence of games ($1 \leq n \leq 10^5$).

The second line contains n space-separated integers a_i. If $a_i = 1$, then the i-th serve was won by Petya, if $a_i = 2$, then the i-th serve was won by Gena.

It is not guaranteed that at least one option for numbers s and t corresponds to the given record.

Output

In the first line print a single number k — the number of options for numbers s and t.

In each of the following k lines print two integers s_i and t_i — the option for numbers s and t. Print the options in the order of increasing s_i, and for equal s_i — in the order of increasing t_i.

Examples

input
5
1 2 1 2 1

output
2
1 3
3 1

input
4
1 1 1 1

output
3
1 4
2 2
4 1

input

```
4
1 2 1 2
```

output

```
0
```

 input

```
8
2 1 2 1 1 1 1 1
```

output

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