Problem I: Intrinsic Interval

Time limit: 3 s
Memory limit: 512 MiB

Given a permutation π of integers 1 through n, an *interval* in π is a consecutive subsequence consisting of consecutive numbers. More precisely, for indices a and b where $1 \leq a \leq b \leq n$, the subsequence $\pi_a^b = (\pi_a, \pi_{a+1}, \ldots, \pi_b)$ is an interval if sorting it would yield a sequence of consecutive integers. For example, in permutation $\pi = (3, 1, 7, 5, 6, 4, 2)$, the subsequence π_3^6 is an interval (it contains the numbers 4 through 7) while π_1^3 is not.

For a subsequence π_x^y its *intrinsic interval* is any interval π_a^b that contains the given subsequence ($a \leq x \leq y \leq b$) and that is, additionally, as short as possible. Of course, the *length* of an interval is defined as the number of elements it contains.

Given a permutation π and m of its subsequences, find some intrinsic interval for each subsequence.

Input

The first line contains an integer n ($1 \leq n \leq 100\,000$) — the size of the permutation π. The following line contains n different integers $\pi_1, \pi_2, \ldots, \pi_n$ ($1 \leq \pi_j \leq n$) — the permutation itself.

The following line contains an integer m ($1 \leq m \leq 100\,000$) — the number of subsequences. The j-th of the following m lines contains integers x_j and y_j ($1 \leq x_j \leq y_j \leq n$) — the endpoints of the j-th subsequence.

Output

Output m lines. The j-th line should contain two integers a_j and b_j where $1 \leq a_j \leq b_j \leq n$ — the endpoints of some intrinsic interval of the j-th subsequence $\pi_{x_j}^{y_j}$.

Example

input

```
7
3 1 7 5 6 4 2
3
3 6
7 7
1 3
```

output

```
3 6
7 7
1 7
```

input

```
10
2 1 4 3 5 6 7 10 8 9
5
2 3
3 7
4 7
4 8
7 8
```

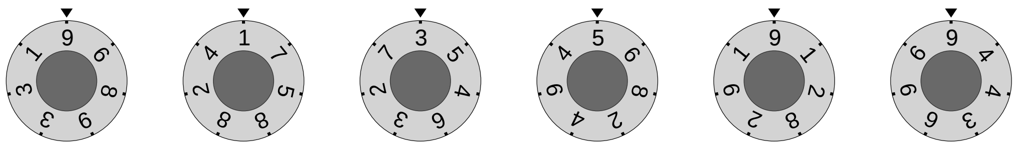
output

```
1 4
3 7
3 7
3 10
7 10
```

Problem K: Kitchen Knobs

Time limit: 3 s
Memory limit: 512 MiB

You are cooking on a gigantic stove at a large fast-food restaurant. The stove contains n heating elements arranged in a line and numbered with integers 1 through n left to right. Each element is operated by its *control knob*. The knobs are a bit unusual: each knob is marked with seven non-zero digits evenly distributed around a circle. The *power* of the heating element is equal to the positive integer obtained by reading the digits on its control knob clockwise starting from the top of the knob.



Initial positions of the control knobs in the first example input below.

In a single step, you can rotate one or more *consecutive* knobs by any number of positions in any direction. However, all knobs rotated in one step need to be rotated by the same number of positions in the same direction.

Find the smallest number of steps needed to set all the heating elements to maximal possible power.

Input

The first line contains an integer n ($1 \leq n \leq 501$) — the number of heating elements. The j-th of the following n lines contains an integer x_j — the initial power of the j-th heating element. Each x_j consists of exactly seven non-zero digits.

Output

Output a single integer — the minimal number of steps needed.

Example

input

```
6
9689331
1758824
3546327
5682494
9128291
9443696
```

output

```
3
```

input

```
7
5941186
3871463
8156346
9925977
8836125
9999999
5987743
```

output

```
2
```

In the first example, one of the ways to achieve maximal possible power is: rotate knobs 2 through 3 by 3 positions in the counterclockwise direction, rotate knob 3 by 3 positions in the counterclockwise direction, and rotate knobs 4 through 6 by 2 positions in the clockwise direction.