A. Borya and Hanabi

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Have you ever played Hanabi? If not, then you've got to try it out! This problem deals with a simplified version of the game.

Overall, the game has 25 types of cards (5 distinct colors and 5 distinct values). Borya is holding n cards. The game is somewhat complicated by the fact that everybody sees Borya's cards except for Borya himself. Borya knows which cards he has but he knows nothing about the order they lie in. Note that Borya can have multiple identical cards (and for each of the 25 types of cards he knows exactly how many cards of this type he has).

The aim of the other players is to achieve the state when Borya knows the color and number value of each of his cards. For that, other players can give him hints. The hints can be of two types: color hints and value hints.

A color hint goes like that: a player names some color and points at all the cards of this color.

Similarly goes the value hint. A player names some value and points at all the cards that contain the value.

Determine what minimum number of hints the other players should make for Borya to be certain about each card's color and value.

Input

The first line contains integer n $(1 \le n \le 100)$ — the number of Borya's cards. The next line contains the descriptions of n cards. The description of each card consists of exactly two characters. The first character shows the color (overall this position can contain five distinct letters — R, G, B, Y, W). The second character shows the card's value (a digit from 1 to 5). Borya doesn't know exact order of the cards they lie in.

Output

Print a single integer — the minimum number of hints that the other players should make.

Sample test(s)

input
```
2
G3 G3
```
output
```
0
```

input
```
4
G4 R4 R3 B3
```
output
```
2
```

input
```
5
B1 Y1 W1 G1 R1
```
output
```
4
```

Note

In the first sample Borya already knows for each card that it is a green three.

In the second sample we can show all fours and all red cards.

In the third sample you need to make hints about any four colors.

B. Andrey and Problem

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Andrey needs one more problem to conduct a programming contest. He has n friends who are always willing to help. He can ask some of them to come up with a contest problem. Andrey knows one value for each of his fiends — the probability that this friend will come up with a problem if Andrey asks him.

Help Andrey choose people to ask. As he needs only one problem, Andrey is going to be really upset if no one comes up with a problem or if he gets more than one problem from his friends. You need to choose such a set of people that maximizes the chances of Andrey not getting upset.

Input

The first line contains a single integer n $(1 \le n \le 100)$ — the number of Andrey's friends. The second line contains n real numbers p_i $(0.0 \le p_i \le 1.0)$ — the probability that the i-th friend can come up with a problem. The probabilities are given with at most 6 digits after decimal point.

Output

Print a single real number — the probability that Andrey won't get upset at the optimal choice of friends. The answer will be considered valid if it differs from the correct one by at most 10^{-9}.

Sample test(s)

input

```
4
0.1 0.2 0.3 0.8
```

output

```
0.800000000000
```

input

```
2
0.1 0.2
```

output

```
0.260000000000
```

Note

In the first sample the best strategy for Andrey is to ask only one of his friends, the most reliable one.

In the second sample the best strategy for Andrey is to ask all of his friends to come up with a problem. Then the probability that he will get exactly one problem is $0.1 \cdot 0.8 + 0.9 \cdot 0.2 = 0.26$.

C. Artem and Array

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Artem has an array of n positive integers. Artem decided to play with it. The game consists of n moves. Each move goes like this. Artem chooses some element of the array and removes it. For that, he gets $min(a, b)$ points, where a and b are numbers that were adjacent with the removed number. If the number doesn't have an adjacent number to the left or right, Artem doesn't get any points.

After the element is removed, the two parts of the array glue together resulting in the new array that Artem continues playing with. Borya wondered what maximum total number of points Artem can get as he plays this game.

Input

The first line contains a single integer n $(1 \le n \le 5 \cdot 10^5)$ — the number of elements in the array. The next line contains n integers a_i $(1 \le a_i \le 10^6)$ — the values of the array elements.

Output

In a single line print a single integer — the maximum number of points Artem can get.

Sample test(s)

input

```
5
3 1 5 2 6
```

output

```
11
```

input

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

output

```
6
```

input

```
5
1 100 101 100 1
```

output

```
102
```

D. Adam and Tree

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

When Adam gets a rooted tree (connected non-directed graph without cycles), he immediately starts coloring it. More formally, he assigns a color to each edge of the tree so that it meets the following two conditions:

- There is no vertex that has more than two incident edges painted the same color.
- For any two vertexes that have incident edges painted the same color (say, c), the path between them consists of the edges of the color c.

Not all tree paintings are equally good for Adam. Let's consider the path from some vertex to the root. Let's call the number of distinct colors on this path the cost of the vertex. The cost of the tree's coloring will be the maximum cost among all the vertexes. Help Adam determine the minimum possible cost of painting the tree.

Initially, Adam's tree consists of a single vertex that has number one and is the root. In one move Adam adds a new vertex to the already existing one, the new vertex gets the number equal to the minimum positive available integer. After each operation you need to calculate the minimum cost of coloring the resulting tree.

Input

The first line contains integer n $(1 \le n \le 10^6)$ — the number of times a new vertex is added. The second line contains n numbers p_i $(1 \le p_i \le i)$ — the numbers of the vertexes to which we add another vertex.

Output

Print n integers — the minimum costs of the tree painting after each addition.

Sample test(s)

input

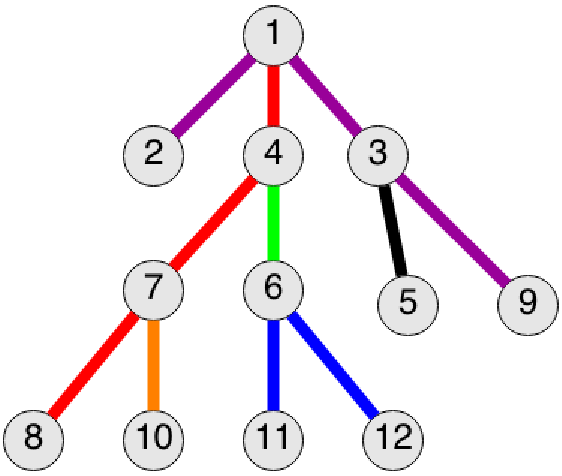
```
11
1 1 1 3 4 4 7 3 7 6 6
```

output

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

Note

The figure below shows one of the possible variants to paint a tree from the sample at the last moment. The cost of the vertexes with numbers 11 and 12 equals 3.



E. Gena and Second Distance

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Gena doesn't like geometry, so he asks you to solve this problem for him.

A rectangle with sides parallel to coordinate axes contains n dots. Let's consider some point of the plane. Let's count the distances from this point to the given n points. Let's sort these numbers in the non-decreasing order. We'll call the beauty of the point the second element of this array. If there are two mimimum elements in this array, the beaty will be equal to this minimum.

Find the maximum beauty of a point inside the given rectangle.

Input

The first line contains three integers w, h, n ($1 \le w, h \le 10^6$, $2 \le n \le 1000$) — the lengths of the rectangle sides and the number of points. Next n lines contain two integers x_i, y_i ($0 \le x_i \le w$, $0 \le y_i \le h$) each — the coordinates of a point. It is possible that it will be coincident points.

Output

Print a single number — the maximum beauty of a point with the absolute or relative error of at most 10^{-9}.

Sample test(s)

input

```
5 5 4
0 0
5 0
0 5
5 5
```

output

```
4.99999999941792340
```

input

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

output

```
5.65685424744772010
```

Note

The point which beauty we need to find must have coordinates (x, y), where $0 \le x \le w$, $0 \le y \le h$. Some of the n points can coincide.