CODEFORCES β
Sponsored by Telegram

Codeforces Round #485 (Div. 2)

A. Infinity Gauntlet

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You took a peek on Thanos wearing Infinity Gauntlet. In the Gauntlet there is a place for six Infinity Gems:

- the `Power` Gem of `purple` color,
- the `Time` Gem of `green` color,
- the `Space` Gem of `blue` color,
- the `Soul` Gem of `orange` color,
- the `Reality` Gem of `red` color,
- the `Mind` Gem of `yellow` color.

Using colors of Gems you saw in the Gauntlet determine the names of **absent** Gems.

Input

In the first line of input there is one integer $$$n$$$ ($$$0 \le n \le 6$$$) — the number of Gems in Infinity Gauntlet.

In next $$$n$$$ lines there are colors of Gems you saw. Words used for colors are: `purple`, `green`, `blue`, `orange`, `red`, `yellow`. It is guaranteed that all the colors are **distinct**. All colors are given in lowercase English letters.

Output

In the first line output one integer $$$m$$$ ($$$0 \le m \le 6$$$) — the number of absent Gems.

Then in $$$m$$$ lines print the names of absent Gems, each on its own line. Words used for names are: `Power`, `Time`, `Space`, `Soul`, `Reality`, `Mind`. Names can be printed in any order. Keep the first letter uppercase, others lowercase.

Examples

input

```
4
red
purple
yellow
orange
```

output

```
2
Space
Time
```

input

```
0
```

output

```
6
Time
Mind
Soul
Power
Reality
Space
```

Note

In the first sample Thanos already has `Reality`, `Power`, `Mind` and `Soul` Gems, so he needs two more: `Time` and `Space`.

In the second sample Thanos doesn't have any Gems, so he needs all six.

B. High School: Become Human

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Year 2118. Androids are in mass production for decades now, and they do all the work for humans. But androids have to go to school to be able to solve creative tasks. Just like humans before.

It turns out that high school struggles are not gone. If someone is not like others, he is bullied. Vasya-8800 is an economy-class android which is produced by a little-known company. His design is not perfect, his characteristics also could be better. So he is bullied by other androids.

One of the popular pranks on Vasya is to force him to compare $$$x^y$$$ with $$$y^x$$$. Other androids can do it in milliseconds while Vasya's memory is too small to store such big numbers.

Please help Vasya! Write a fast program to compare $$$x^y$$$ with $$$y^x$$$ for Vasya, maybe then other androids will respect him.

Input

On the only line of input there are two integers $$$x$$$ and $$$y$$$ ($$$1 \le x, y \le 10^{9}$$$).

Output

If $$$x^y < y^x$$$, then print '<' (without quotes). If $$$x^y > y^x$$$, then print '>' (without quotes). If $$$x^y = y^x$$$, then print '=' (without quotes).

Examples

input

```
5 8
```

output

```
>
```

input

```
10 3
```

output

```
<
```

input

```
6 6
```

output

```
=
```

Note

In the first example $$$5^8 = 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 = 390625$$$, and $$$8^5 = 8 \cdot 8 \cdot 8 \cdot 8 \cdot 8 = 32768$$$. So you should print '>'.

In the second example $$$10^3 = 1000 < 3^{10} = 59049$$$.

In the third example $$$6^6 = 46656 = 6^6$$$.

C. Three displays

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

It is the middle of 2018 and Maria Stepanovna, who lives outside Krasnokamensk (a town in Zabaikalsky region), wants to rent three displays to highlight an important problem.

There are $$$n$$$ displays placed along a road, and the $$$i$$$-th of them can display a text with font size $$$s_i$$$ only. Maria Stepanovna wants to rent such three displays with indices $$$i < j < k$$$ that the font size increases if you move along the road in a particular direction. Namely, the condition $$$s_i < s_j < s_k$$$ should be held.

The rent cost is for the $$$i$$$-th display is $$$c_i$$$. Please determine the smallest cost Maria Stepanovna should pay.

Input

The first line contains a single integer $$$n$$$ ($$$3 \le n \le 3\,000$$$) — the number of displays.

The second line contains $$$n$$$ integers $$$s_1, s_2, \ldots, s_n$$$ ($$$1 \le s_i \le 10^9$$$) — the font sizes on the displays in the order they stand along the road.

The third line contains $$$n$$$ integers $$$c_1, c_2, \ldots, c_n$$$ ($$$1 \le c_i \le 10^8$$$) — the rent costs for each display.

Output

If there are no three displays that satisfy the criteria, print `-1`. Otherwise print a single integer — the minimum total rent cost of three displays with indices $$$i < j < k$$$ such that $$$s_i < s_j < s_k$$$.

Examples

input

```
5
2 4 5 4 10
40 30 20 10 40
```

output

```
90
```

input

```
3
100 101 100
2 4 5
```

output

```
-1
```

input

```
10
1 2 3 4 5 6 7 8 9 10
10 13 11 14 15 12 13 13 18 13
```

output

```
33
```

Note

In the first example you can, for example, choose displays 1, 4 and 5, because $s_1 < s_4 < s_5$ ($2 < 4 < 10$), and the rent cost is $40 + 10 + 40 = 90$.

In the second example you can't select a valid triple of indices, so the answer is `-1`.

D. Fair

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

Some company is going to hold a fair in Byteland. There are n towns in Byteland and m two-way roads between towns. Of course, you can reach any town from any other town using roads.

There are k types of goods produced in Byteland and every town produces only one type. To hold a fair you have to bring at least s different types of goods. It costs $d(u,v)$ coins to bring goods from town u to town v where $d(u,v)$ is the length of the shortest path from u to v. Length of a path is the number of roads in this path.

The organizers will cover all travel expenses but they can choose the towns to bring goods from. Now they want to calculate minimum expenses to hold a fair in each of n towns.

Input

There are 4 integers n, m, k, s in the first line of input ($1 \le n \le 10^{5}$, $0 \le m \le 10^{5}$, $1 \le s \le k \le min(n, 100)$) — the number of towns, the number of roads, the number of different types of goods, the number of different types of goods necessary to hold a fair.

In the next line there are n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_{i} \le k$), where a_i is the type of goods produced in the i-th town. It is guaranteed that all integers between 1 and k occur at least once among integers a_{i}.

In the next m lines roads are described. Each road is described by two integers u v ($1 \le u, v \le n$, $u \ne v$) — the towns connected by this road. It is guaranteed that there is no more than one road between every two towns. It is guaranteed that you can go from any town to any other town via roads.

Output

Print n numbers, the i-th of them is the minimum number of coins you need to spend on travel expenses to hold a fair in town i. Separate numbers with spaces.

Examples

input

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

output

```
2 2 2 2 3
```

input

```
7 6 3 2
1 2 3 3 2 2 1
1 2
2 3
```

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

output

```
1 1 1 2 2 1 1
```

Note

Let's look at the first sample.

To hold a fair in town $$$1$$$ you can bring goods from towns $$$1$$$ ($$$0$$$ coins), $$$2$$$ ($$$1$$$ coin) and $$$4$$$ ($$$1$$$ coin). Total numbers of coins is $$$2$$$.

Town $$$2$$$: Goods from towns $$$2$$$ ($$$0$$$), $$$1$$$ ($$$1$$$), $$$3$$$ ($$$1$$$). Sum equals $$$2$$$.

Town $$$3$$$: Goods from towns $$$3$$$ ($$$0$$$), $$$2$$$ ($$$1$$$), $$$4$$$ ($$$1$$$). Sum equals $$$2$$$.

Town $$$4$$$: Goods from towns $$$4$$$ ($$$0$$$), $$$1$$$ ($$$1$$$), $$$5$$$ ($$$1$$$). Sum equals $$$2$$$.

Town $$$5$$$: Goods from towns $$$5$$$ ($$$0$$$), $$$4$$$ ($$$1$$$), $$$3$$$ ($$$2$$$). Sum equals $$$3$$$.

E. Petr and Permutations

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

Petr likes to come up with problems about randomly generated data. This time problem is about random permutation. He decided to generate a random permutation this way: he takes identity permutation of numbers from $$$1$$$ to $$$n$$$ and then $$$3n$$$ times takes a random pair of different elements and swaps them. Alex envies Petr and tries to imitate him in all kind of things. Alex has also come up with a problem about random permutation. He generates a random permutation just like Petr but swaps elements $$$7n+1$$$ times instead of $$$3n$$$ times. Because it is more random, OK?!

You somehow get a test from one of these problems and now you want to know from which one.

Input

In the first line of input there is one integer $$$n$$$ ($$$10^{3} \le n \le 10^{6}$$$).

In the second line there are $$$n$$$ distinct integers between $$$1$$$ and $$$n$$$ — the permutation of size $$$n$$$ from the test.

It is guaranteed that all tests except for sample are generated this way: First we choose $$$n$$$ — the size of the permutation. Then we randomly choose a method to generate a permutation — the one of Petr or the one of Alex. Then we generate a permutation using chosen method.

Output

If the test is generated via Petr's method print "`Petr`" (without quotes). If the test is generated via Alex's method print "`Um_nik`" (without quotes).

Example

input

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

output

```
Petr
```

Note

Please note that the sample is not a valid test (because of limitations for $$$n$$$) and is given only to illustrate input/output format. Your program **still has to print correct answer to this test** to get AC.

Due to randomness of input hacks in this problem are forbidden.

F. AND Graph

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a set of size $$$m$$$ with integer elements between $$$0$$$ and $$$2^{n}-1$$$ inclusive. Let's build an undirected graph on these integers in the following way: connect two integers $$$x$$$ and $$$y$$$ with an edge if and only if $$$x \& y = 0$$$. Here $$$\&$$$ is the bitwise AND operation. Count the number of connected components in that graph.

Input

In the first line of input there are two integers $$$n$$$ and $$$m$$$ ($$$0 \le n \le 22$$$, $$$1 \le m \le 2^{n}$$$).

In the second line there are $$$m$$$ integers $$$a_1, a_2, \ldots, a_m$$$ ($$$0 \le a_{i} < 2^{n}$$$) — the elements of the set. All $$$a_{i}$$$

are distinct.

Output

Print the number of connected components.

Examples

input

```
2 3
1 2 3
```

output

```
2
```

input

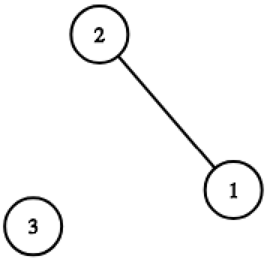
```
5 5
5 19 10 20 12
```

output

```
2
```

Note

Graph from first sample:



Graph from second sample: