XII Concurso Anual de Programación Donald Knuth

A. Alan and Figuratively Speaking

1 second, 256 megabytes

Some time ago, certain passage circulated the social media. The title of that passage was F1gur471v31y 5p34k1ng and it stated the following:

7H15 M3554G3 53RV35 7O PROV3 H0W 0UR M1ND5 C4N D0 4M4Z1NG 7H1NG5! 1MPR3551V3 7H1NG5! IN 7HE B3G1NN1NG 17 WA5 H4RD BU7 N0W, 0N 7H15 L1N3, Y0UR M1ND 1S R34D1NG 17 4U70M47C4LLY W17H 0U7 3V3N 7H1NKING 4B0U7 17.

When Alan read about this message, he found it specially interesting. Now, he wants to create his own passage, using a message consisting of English letters and replacing some letters with digits.

The replacement process consists in replacing all of the letters **O** with the digit **0**, **I** with **1**, **Z** with **2**, **E** with **3**, **A** with **4**, **S** with **5**, **G** with **6**, **T** with **7** and **B** with **8**. The letters must be replaced no matter if they are lowercase or uppercase. For example, 'String' should be replaced with '57r1n6'.

Help Alan to replace the letters of his message. To simplify the process, the message contains underscores ('_') instead of spaces.

Input

The input contains a single string consisting of at most 50 characters. Each character may be an English letter (lowercase or uppercase) or an underscore ('_').

Output

Print the answer.

Scoring

There are no subtasks for this problem.

input
```
Concurso_Anual_de_Programacion_Donald_Knuth
```
output
```
C0ncur50_4nu4l_d3_Pr06r4m4c10n_D0n4ld_Knu7h
```

input
```
Huron_Legendario
```
output
```
Hur0n_L363nd4r10
```

input
```
String
```
output
```
57r1n6
```

B. Beautiful Necklace

2 seconds, 256 megabytes

Some days ago Alice received a thread with beads as a birthday present. The color of each bead is one out of four possible colors: red, green, blue or yellow. The thread can be represented by a string consisting of 'R', 'G', 'B' and 'Y', where the i-th character represents the color of the i-th bead in the thread. She wants to cut some contiguous breads from the thread to make a necklace.

Alice thinks that the beauty of a necklace is inversely proportional to the difference between the occurrences of the most frequent and the least frequent color in the necklace. So, she wants to select a contiguous sequence that minimizes this difference. She also likes long necklaces, so, if there are multiple contiguous sequences that minimizes this difference, she will choose the longest one.

For example, if the thread is 'RGGGBBBY', she can choose the contiguous subsequence 'RG', because the most frequent color occurs one time and the least frequent color occurs zero times, so the difference is 1, which is minimum. There are multiple contiguous subsequences that gives the difference of 1, but the longest one has length 2. Note that if one of the four colors doesn't occur at all, then its frequency is said to be 0.

Help Alice to find the best contiguous subsequence for her necklace.

Input

The input contains a single string S — the description of the necklace. It is guaranteed that the string contains only the characters 'R', 'G', 'B' and 'Y'.

Output

In the first line print two space-separated integers — the minimum difference between the occurrences of the the most frequent and the least frequent color, and the maximum length of a contiguous subsequence that minimizes this difference.

In the second line print two space-separated integers — the index of the start and the end of such a contiguous subsequence. The indexes are 1-based. If there are many suitable indices, print any of them.

Scoring

Subtask #1 (10 points): $1 \leq |S| \leq 300$

Subtask #2 (20 points): $1 \leq |S| \leq 4000$

Subtask #3 (70 points): $1 \leq |S| \leq 10^5$

input
```
RGBY
```
output
```
0 4
1 4
```

input
```
RGB
```
output
```
1 3
1 3
```

input
```
RGGGBBBY
```
output
```
1 2
1 2
```

The following output is also acceptable for the third example:

$1\ 2$

$4\ 5$

C. Conquering Paths

1.5 seconds, 256 megabytes

El Capitan is a famous pirate who some time ago conquered some cities in Treeland. But he is not done, and this time he returned to Treeland to conquer more cities.

Treeland can be described as a connected graph consisting of N cities and $N-1$ roads. Each city has a treasure type, which can be gold, rubies, diamonds, etc. The treasure type of the i-th city is identified by an integer a_i. El Capitan will conquer the cities that lay in some simple path, but before that, he wants to know some information about the treasures in simple paths. Help El Capitan and process Q queries which can be of two types:

1. Given u and v, let $dist(u, v)$ be the number of nodes in the simple path between u and v, answer if there is a treasure type that occurs **strictly** more than $\frac{dist(u,v)}{2}$ times in this path. If so, print that treasure type, or say that there is no such treasure type.
2. Given u and k, change the treasure type in the u-th city to k.

Recall that a simple path is a path where each vertex is visited exactly once.

Input
The first line contains an integer N.

The second line contains N integers a_1, a_2, ...,a_N ($1 \le a_i \le 10^5$).

The following $N-1$ lines contains two integers u and v, indicating that there is a road connecting cities u and v.

The next line contains an integer Q.

The next Q lines contain the description of the queries. Each line starts with an integer t. If $t = 1$, then it is a query of the first type and two integers u and v ($1 \le u, v \le N$) follow. If $t = 2$, then it is a query of the second type and two integers u and k follow ($1 \le u, v \le N$ and $1 \le k \le 10^5$).

It is guaranteed that there is a least a query of the first type.

Output
For each query of the first type, print a line with the treasure type that occurs more than $\frac{dist(u,v)}{2}$ times in the simple path between u and v, or with -1 if there is no such treasure type.

Scoring
Subtask #1 (30 points): $1 \le N, Q \le 1000$

Subtask #2 (70 points): $1 \le N, Q \le 10^5$

input
```
4
1 1 2 2
1 2
1 3
1 4
7
1 2 4
1 3 4
1 1 2
2 1 2
1 2 4
1 3 4
1 1 2
```

output
```
1
2
1
2
2
-1
```

D. Disaster rounding

1 second, 256 megabytes

You are given a positive integer n. Compute the following sum:

$$\sum_{k=1}^{n} \lfloor \sqrt{k} \rceil$$

Where $\lfloor x \rceil$ is the nearest integer to x (the round function).

Since the answer can be pretty big, find it modulo $10^9 + 7$.

Input
The first and only line contains the positive integer n.

Output
Print a single line containing the answer.

Scoring

- Subtask #1 (10 points): $n \le 10^6$
- Subtask #2 (30 points): $n \le 10^{14}$
- Subtask #3 (60 points): $n \le 10^{18}$

input
```
3
```
output
```
4
```

input
```
4
```
output
```
6
```

input
```
6
```
output
```
10
```

input
```
19
```
output
```
56
```

E. Elisa and blessed numbers

0.75 seconds, 128 megabytes

You have to find the n^{th} positive integer such that, when represented in base 10 with no leading zeroes, it doesn't contain any substring that matches with a string from a set of m forbidden strings.

Input
The first line contains the positive integer m ($1 \le m \le 100$).

Then, m lines follow: $s_1, s_2, \ldots, s_m$ ($|s_i| \leq 18$), the strings of the forbidden set, consisting only of digits from 0 to 9. They don't contain any leading zeroes.

Finally, the last line contains the positive integer n.

Output

Output one positive integer, the required number. It's guaranteed that the answer exists and won't exceed $10^{19} + 19$.

Scoring

- Subtask #1 (25 points): $n \leq 500$
- Subtask #2 (75 points): $n < 10^{19}$

input
```
1
4
9
```
output
```
10
```

input
```
2
1
6
54
```
output
```
88
```

In the first case, the first valid numbers are: $1, 2, 3, 5, 6, 7, 8, 9, 10, 11, \ldots$, and the 9^{th} of them is 10.

F. Social Distancing

1 second, 256 megabytes

Given to the Covid-19 pandemic situation that the world is living right now, there have been different measures and programs that governments have already put in place.

For example, the Mexico government has launched the program called "Mantener la sana distancia". Mexico government urges public to keep distance over coronavirus.

People started following the guidelines, but for some reason they still want to go outside and take fresh air. Due to a lot of people going to the streets and not following the measures to reduce the number of coronavirus cases, the Mexican government wants to propose a new program called "Toma aire fresco manteniendo la sana distancia".

For this new project, people will be allowed to be on the streets in an assigned circular area inside the city if they stay inside the assigned area. This area is defined by a point p_i and a radious of **150 centimeters** around the assigned point location.

But there was a problem with this plan! All people went to the street at the same time without taking the necessary measures.

The president decided to cancel the permit of staying outside for some people because it could be very dangerous for them to be exposed to the coronavirus disease. The president wants the maximum number of people to be happy (people become happy when they take fresh air outside in their assigned areas) and he needs your help to solve this problem.

Given the size of the country represented by a grid of size N x M, T people and their assigned areas locations defined by $p_i = (x_i, y_i)$ and a radious of **150 centimeters** where they are allowed to take fresh air, you need to print the maximum number of people that can be outside taking fresh air such that every person in the country stays only in their assigned areas.

In other words, you need to print the maximum number of people that can be in the grid, such that nobody is able to enter to other person's assigned area while they take a walk in their assigned spots.

It is not allowed by the government if a person is exactly at the border of other people's spot.

All test cases were generated with a uniform distribution of probability.

Input

The first line of the input will contain two numbers N and M - Representing the size of the country.

The second line of the input will contain a number T - Representing the number of people that live in the country.

Next T lines will contain two numbers x_i and y_i $(1 \leq x_i \leq N, 1 \leq y_i \leq M)$ - Representing the assigned spot for each person where they can take fresh air.

It is guaranteed that not two persons will be assigned with the same point location to be able to take fresh air.

Output

Output one positive integer, the maximum number of people that can be outside at the same time taking fresh air in their assigned point locations.

Scoring

Subtask #1 (30 Points):

$$N = 1, 1 \leq M \leq 10^5$$

$$1 \leq T \leq M$$

Subtask #2 (70 Points):

$$N, M = 10^5$$

$$1 \leq T \leq 600$$

input
```
1 1500
5
1 150
1 350
1 550
1 750
1 1000
```
output
```
3
```

input
```
1 400
2
1 50
1 400
```
output
```
2
```

input

```
100000 100000
5
500 500
800 500
500 200
200 500
500 800
```

output

```
4
```

G. Game of Numbers

2 seconds, 256 megabytes

Abraham and Pachita are playing a cooperative game called *Game of Numbers*. At the beginning of the game, a list of n positive integers is given to the players. Then, the game is played in two phases:

1. During the first phase, the players can reorder the digits within each numbers of the list as they want. It is **not** allowed to move digits between different numbers. For example, if the list consists of two numbers '356' and '842', then they can reorder the digits within the numbers to transform them into '653' and '248'.
2. After that, the players concatenate the numbers of the list in any order to get a greater number. The digits can not be reordered after concatenate the numbers of the list. For example, if after the first phase the numbers of the list are '653' and '248', then the players can get '653248' or '248653' after concatenate them.

The goal of the game is to get the maximum possible number after concatenate the individual numbers of the list. Help Abraham and Pachita to determine what is the maximum number that they can get.

Input

The first line contains an integer n — the number of integers in the list

Then n lines follow. The $(i+1)$-th line contains a positive integer a_i — the i-th number in the list.

Output

Print a single line with the answer.

Note that the answer might not fit in a 64 bit integer.

Scoring

Subtask #1 (15 points):

$1 \le n \le 10^5$

$1 \le a_i < 10.$

Subtask #2 (25 points):

$1 \le n < 10$

$1 \le a_i < 10^{18}$

Subtask #3 (60 points):

$1 \le n \le 10^5$

$1 \le a_i < 10^{18}$

input

```
4
2
3
3
1
```

output

```
3321
```

input

```
2
10
19
```

output

```
9110
```

input

```
2
5
5155
```

output

```
55551
```

In the second example, a player can reorder the digits in the second number of the list to get 91, and then concatenate 91 and 10 to get 9110.

H. Huron Sorting

1 second, 128 megabytes

Norman likes to play mobile games. Some days ago, he started to play a new game called *Huron Sorting*.

The game rules are simple: the player is given a permutation (to see the definition of a permutation, please look at the notes section below) of length n and he has to sort it applying some operations on it. An operation consists of choosing two integers i and j such that $i \neq j$ and swap the elements p_i and p_j in the permutation. After performing an operation, the player can choose to continue applying operations or to stop the game. The player wins if the permutation is sorted in ascending order after stopping the game.

Norman has k seconds to finish the game before his next class. He is so skilled as to apply one operation per second and doesn't want to have free time after stopping the game. So, he needs your help to determine if the given permutation can be sorted in ascending order after applying exactly k operations.

Input

The input consists of several test cases. The first line contain an integer T $(1 \le T \le 50000)$ — number of test cases in input. The following $2T$ lines contain description of test cases.

The first line of a test case contains two integers n and k ($2 \le n \le 10^5$) — the length of permutation p and number of second before the next class of Norman.

The second line of a test case contains n integers $p_1, p_2, \ldots, p_n$ — elements in permutation p.

It is guaranteed that the sum of n over all test cases doesn't exceed 10^6.

Output

For each test case, print 'Yes' if it is possible to sort permutation p after applying exactly k operations; otherwise, print 'No'.

Scoring

Subtask #1 (8 points): $k = 0$

Subtask #2 (10 points): $k = 1$

Subtask #3 (10 points): $k = 2$

Subtask #4 (15 points): $k = 3$

Subtask #5 (57 points): $0 \le k \le 10^5$

input

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

output

```
Yes
No
```

A permutation is an array consisting of n distinct digits from 1 to n. For example, $[3, 4, 2, 1, 5]$ is a permutation of 5 integers, but $[1, 2, 3, 5, 2]$ and $[1, 2, 3, 5, 6]$ are not.