Croatian Open Competition in Informatics

Round 1, October 5th 2024

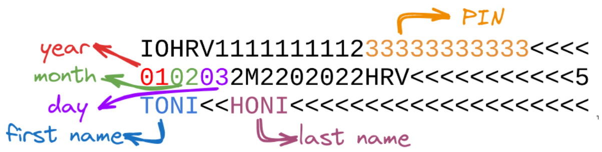
Tasks

Task	Time limit	Memory limit	Score
Osobna	1 second	512 MiB	50
Skokovi	5 seconds	512 MiB	70
Hijerarhija	3 seconds	512 MiB	90
Učiteljica	5 seconds	512 MiB	120
Zbunjenost	5 seconds	512 MiB	120
Total			450

Task Osobna

Toni, the new chairperson of COCI (in Croatian: *HONI*), loves HONI so much that he decided to change his last name to match the competition's name. From now on, his name is Toni Honi! Excited about the change, Toni eagerly awaited his new personal ID card. When it finally arrived, he couldn't stop staring at it. He was especially fascinated by the three lines of characters on the back of the card.

It didn't take him long to realize that these lines contain some of his personal details: his first name, last name, birth date, and personal identification number (PIN). He copied the lines from his ID onto a piece of paper and identified the details he could recognize:



In the first line, which consists of 5 letters, 21 digits, and 4 '<' characters, Toni discovered his PIN — the last 11 digits of the line.

In the second line, he identified his birth date. The first two digits represent the year, the next two the month, and the final two digits the day of birth. If the two-digit year is less than or equal to `24`, it means the person was born in the 21st century (so the full year starts with `20`). Otherwise, the person was born in the 20th century (so the full year starts with `19`).

In the third line, Toni found his first name, followed by two '<' characters, and then his last name. The rest of the line was padded with '<' characters.

Toni found these discoveries fascinating and now wants you to join the fun — for the three lines on the back of the ID card, extract and print the person's first name, last name, birth date, and personal identification number (PIN).

Input

The input consists of three lines of characters, each of length 30. The characters that can appear are uppercase English letters, digits, and the character `'<'`. The first name and last name will each consist of exactly one word and will be in uppercase English letters. Both names will fit entirely into the third line.

Output

Output four lines (**even if not all parts of the problem are solved**):

- In the first line, print `'Ime: '` followed by the person's first name, capitalized (only the first letter should be uppercase).
- In the second line, print `'Prezime: '` followed by the person's last name, capitalized (only the first letter should be uppercase).
- In the third line, print `'Datum rodjenja: '` followed by the birth date in the format `DD-MM-YYYY`.
- In the fourth line, print `'OIB: '` followed by the person's personal identification number (PIN).

Side note: "Ime" stands for name, "Prezime" for surname, and "Datum rodjenja" for birth date when translated from Croatian.

Scoring

The correct output of the first, second, or fourth line on all test cases in a subtask carries one-fifth of the points assigned for that subtask, while a correct output of the third line carries two-fifths of the points.

Subtask	Points	Constraints
1	25	Both the first and last name will consist of only one letter.
2	15	The birth year will be between 2000 and 2024.
3	10	No additional constraints.

Examples

input

```
IOHRV111111111233333333333<<<<
0102032M2202022HRV<<<<<<<<<<<5
TONI<<HONI<<<<<<<<<<<<<<<<<<<<
```

output

```
Ime: Toni
Prezime: Honi
Datum rodjenja: 03-02-2001
OIB: 33333333333
```

input

```
IOHRV123456789012345678901<<<<
9803049F2401059HRV<<<<<<<<<<<2
PAJO<<PATAK<<<<<<<<<<<<<<<<<<<
```

output

```
Ime: Pajo
Prezime: Patak
Datum rodjenja: 04-03-1998
OIB: 12345678901
```

Explanation of the first example:
Refer to the illustration in the problem statement.

Task Skokovi

In a land who knows where, in a world who knows when, lives a bee named Maya. Her adventure-filled life is an endless source of ideas for tasks, so we chose one of them.

Maya's friend, a grasshopper named Filip, is preparing for the Olympics in flower jumping. The flowers in the meadow can be represented as a sequence of positive intergers a of length N. The height of each flower is given by the number a_i.

Filip always jumps from left to right. Additionally, since this whole sport is new to him, he cannot jump to a flower whose height differs too much from the one he is currently on. Specifically, from flower i, he can jump to flower j if and only if $i < j$ and $|a_i - a_j| \leq K$, where K is a positive integer given in the input.

Help Maya plan Filip's training by determining which flowers Filip can reach if he starts on the leftmost flower. In other words, for each flower, determine if there is a sequence of jumps that leads to it, starting from the first flower.

Input

The first line contains positive intergers N $(1 \leq N \leq 2 \cdot 10^5)$ and K $(1 \leq K \leq 10^9)$.

The second line contains a sequence of positive intergers a, i.e., N numbers a_i $(1 \leq a_i \leq 10^9)$, representing the heights of the flowers.

Output

In a single line, output N numbers, either 0 or 1, where each number indicates whether the corresponding flower is reachable. The number 0 means that it is not possible to reach that flower, while 1 means the flower is reachable. The first flower is always reachable as Filip starts from there.

Scoring

Subtask	Points	Constraints
1	20	Flower heights are strictly increasing.
2	25	$N \leq 1\,000$
3	25	No additional constraints.

Examples

input

```
5 2
5 4 8 7 2
```

output

```
1 1 0 1 1
```

input

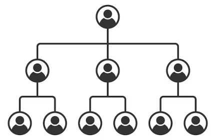
```
5 3
10 15 14 8 9
```

output

```
1 0 0 1 1
```

Clarification of the first example:
Filip can jump directly from the first flower to the second. The third flower is unreachable as Filip cannot jump to it from either the first or the second flower. To reach the fourth flower, Filip can also jump directly from the first flower. For the last flower, Filip needs to jump first to the second flower and then to the last flower.

Task Hijerarhija



Krešimir has started studying corporate structures, including *hierarchies*. He observed employees and their relationships within a company. In this case, we are only looking at *superior–subordinate* relationships, meaning relationships where one employee is directly superior to another employee in the company.

A hierarchy is a structure with N employees and $N-1$ superior–subordinate relationships, where there is one person who is directly or indirectly superior to all employees. In the observed company, there are also N employees and $N-1$ such relationships, but it is not certain whether this is a valid hierarchy or not.

Krešimir has asked you to help answer this question. He has recorded all the data in his notebook. Additionally, in his notebook, he will make Q permanent changes by reversing one superior–subordinate relationship such that the subordinate becomes superior to their former superior. After each such change, it is necessary to answer the same question: is the current state a valid hierarchy?

Input

In the first line, there is a positive integer N $(2 \leq N \leq 3 \cdot 10^5)$.

In the next $N-1$ lines, for each $i = 1, 2, \ldots, N-1$, there is a pair of integers p_i and e_i $(1 \leq p_i, e_i \leq N$, $p_i \neq e_i)$, indicating that p_i is directly superior to e_i.

In the next line, there is a non-negative integer Q $(0 \leq Q \leq 10^6)$.

In the following Q lines, there are pairs a_i, b_i $(1 \leq a_i, b_i \leq N$, $a_i \neq b_i)$. It is guaranteed that at that moment, a_i will either be directly superior to b_i or vice versa.

In the test data, it is **guaranteed** that it will be possible to achieve at least one hierarchy with some sequence of reversals.

Output

In the next $Q+1$ lines, for each of the given scenarios, it is necessary to output whether the current structure is a hierarchy, i.e., `"DA"` if it is, or `"NE"` if it is not (without quotation marks).

Scoring

Subtask	Points	Constraints
1	7	$N \leq 300, Q = 0$
2	12	$N, Q \leq 300$
3	16	$N, Q \leq 1000$
4	15	$Q = 0$
5	23	For each $i = 1, 2, \ldots, N-1$, it will hold that i is directly superior to $i+1$ or vice versa.
6	17	No additional constraints.

Examples

input

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

output

```
DA
DA
DA
DA
```

input

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

output

```
DA
NE
DA
DA
NE
```

Task Učiteljica

In the best school in Varaždin, there was an excellent computer science teacher known for her interesting and unusual ideas. Her name was Lana, and she often gave her students impossible or unsolvable problem. Each student only needed to solve 1 problem during the school year to pass the class with an excellent grade. Students who did not solve any tasks by the end of the year would have to repeat the grade. On the last day of school, she wrote an extremely difficult problem on the board, which went as follows:

Imagine you have a sequence of numbers of length N, and you can remove some elements from the beginning or the end (or both). How many ways can you perform such deletions so that after the deletion, there must be at least 1 number that appears exactly once, at least 1 number that appears exactly twice, $\ldots$, and at least 1 number that appears exactly K times.

A student named Fran, who had not solved any problems so far, quickly said, "I know how to solve this problem." Teacher Lana did not believe Fran and told him, "Write the code in the next 30 minutes, and then I'll believe you. If you don't, you'll have to repeat the grade." Fran doesn't know how to code, so he urgently asked for your help to write the code to solve this task. In his hurry, he forgot to explain his idea for solving the task. Write a code that takes as input the numbers N and K, the sequence of N elements, and solve Lana's question to help Fran.

Input

In the first line, there are 2 positive integers N ($1 \leq N \leq 10^5$) and K ($1 \leq K \leq 4$) from the task description.

In the second line, there are N postive integers a_i ($1 \leq a_i \leq N$), the numbers from the task description.

Output

In the first line, print a whole number, the number of distinct deletions such that the conditions from the task hold. Two deletions are different if there is an index that is not deleted in one, but deleted in the other.

Scoring

Subtask	Points	Constraints
1	20	$N \leq 1000$
2	15	$1 \leq a_i \leq k$ for all $i = 1, 2, \ldots, N$
3	35	$K = 1$
4	50	No additional constraints.

Examples

input

```
3 1
1 2 1
```

output

```
6
```

input

```
6 3
6 5 6 4 5 5
```

output

```
1
```

input

```
6 2
5 4 5 2 6 5
```

output

```
5
```

Clarification of the first example:
The possible sequences after deletion are: [1], [2], [1], [1, 2], [2, 1], [1, 2, 1], and in each of the 6 sequences, there is a number that appears exactly once.

Clarification of the second example:
The only sequence after deletion in which there is at least 1 number that appears exactly once, at least 1 number that appears exactly twice, and at least 1 number that appears exactly three times is the given sequence of N numbers.

Task Zbunjenost

Mr. Malnar decided to spend his summer traveling around the world by flying randomly. After some time, he found himself in an unknown country's capital where streets reminded him of a triangulation! More precisely, the city consists of N interesting locations numbered from 1 to N connected by $2N - 3$ streets. Locations 1, 2, …, N are connected in that order to form a convex polygon with N sides. Remaining $N - 3$ streets connect locations in such a way that no two streets cross, except maybe at their ends.

While walking the streets of this country's capital Mr. Malnar found himself at the location where he started from without visiting any location more than once. A bit confused he came to the realization that this is completely normal and came up with a measure of confusion as the number of simple closed loops. Simple closed walk is a sequence of locations V_1, V_2, …, V_m such that V_i is connected via street with location V_{i+1} for every $i = 1, 2, \ldots, m - 1$ and that location V_m is connected with V_1. Two walks are equivalent if one can be obtained by a cyclic rotation of the other, or by reversing the other. For example, walk $(1, 2, 3, 4)$ is equivalent to the walk $(2, 3, 4, 1)$. Simple closed loop is a set of equivalent walks. Mr. Malnar now asks for your help in calculating the confusion of this city!

Input

In the first line is an integer N $(1 \leq N \leq 2 \cdot 10^5)$, number of interesting locations.

In the next $N - 3$ lines are integers X_i, Y_i $(1 \leq X_i, Y_i \leq N)$, labels of locations connected by i-th street.

Output

In the first and only line output the confusion of the given city modulo $10^9 + 7$.

Scoring

Subtask	Points	Constraints
1	13	$N \leq 15$
2	18	$N \leq 300$
3	34	$N \leq 2000$
4	15	Locations 1 and k are connected for all $k = 3, 4, \ldots, N - 1$.
5	40	No additional constraints.

Examples

input

```
4
1 3
```

output

```
3
```

input

```
5
1 3
3 5
```

output

```
6
```

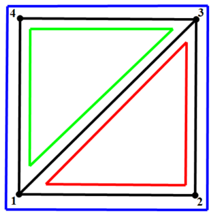
input

```
6
2 4
4 6
6 2
```

output

```
11
```

Clarification of the first example:
On the sketch, each loop is colored in a different color.



Clarification of the second example:
Walks that represent loops are: $(1,2,3)$, $(1,3,5)$, $(3,4,5)$, $(1,2,3,5)$, $(1,3,4,5)$, $(1,2,3,4,5)$.

Clarification of the third example:
Walks that represent loops are: $(1,2,6)$, $(2,3,4)$, $(4,5,6)$, $(2,4,6)$, $(1,2,4,6)$, $(2,3,4,6)$, $(2,4,5,6)$, $(1,2,3,4,6)$, $(2,3,4,5,6)$, $(1,2,4,5,6)$, $(1,2,3,4,5,6)$.