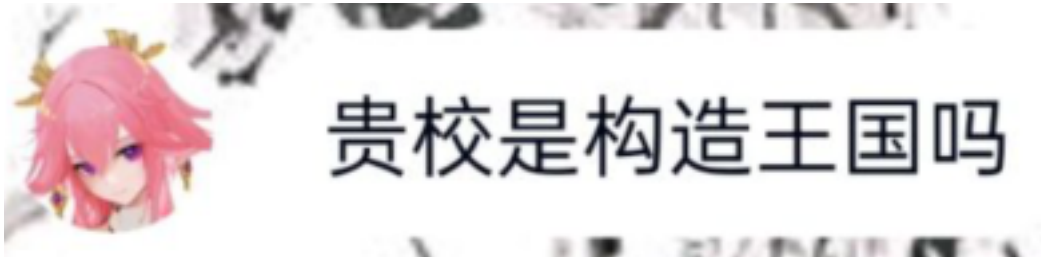


## Problem A. Make SYSU Great Again I

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         256 megabytes

As you know, you guys are participating in CCPC (Chinese Constructive Problem Contest), and no doubt that the proposition school SYSU, as the kingdom of constructive problems, is going to challenge you to solve some related problems.



Given an  $n \times n$  grid, you are asked to fill **each number** in  $[1, k]$  into the grid, and it needs to fulfill the following requirements.

1. Each number from 1 to  $k$  occurs exactly once.
2. Each cell should be filled with at most one number.
3. Each row and column has at least two numbers.
4. For each integer  $i \in [1, n]$ , the greatest common divisor of all numbers in row  $i$  is equal to the greatest common divisor of all numbers in column  $i$ .

It's guaranteed that there must be a valid solution under the restriction of the problem.

### Input

The only line contains two integers  $n$  and  $k$  ( $2 \leq n \leq 2 \times 10^5$ ,  $2 \times n \leq k \leq \min(n^2, 10^6)$ ) — the size of the grid and the numbers you need to fill in.

### Output

You should output  $k$  lines, with line  $i$  consisting of two integers  $x_i, y_i$  representing the rows and columns of the position filled by the number  $i$ , respectively.

If there are multiple solutions, you can output any one.

### Example

standard input	standard output
3 6	1 1 2 2 1 3 2 3 3 1 3 2

## Problem B. Yet Another Subsequence Problem

Input file:            standard input  
Output file:           standard output  
Time limit:            1 second  
Memory limit:         256 megabytes

For any two positive integers  $a$  and  $b$ , define the function by the following C++ code:

```
std::string gen_string(int64_t a, int64_t b) {  
    std::string res;  
    int64_t ia = 0, ib = 0;  
    while (ia + ib < a + b) {  
        if ((__int128)ia * b <= (__int128)ib * a) {  
            res += '0';  
            ia++;  
        } else {  
            res += '1';  
            ib++;  
        }  
    }  
    return res;  
}
```

$ia$  will equal  $a$  and  $ib$  will equal  $b$  when the loop terminates, so this function returns a bitstring of length  $a + b$  with exactly  $a$  zeroes and  $b$  ones. For example,  $gen\_string(4, 10) = 01110110111011$ .

Given the argument of  $A, B$ , you should calculate the number of different subsequences of  $gen\_string(A, B)$ , and print it modulo 998244353.

**NOTE:** A sequence  $a$  is a subsequence of a string  $b$  if  $a$  can be obtained from  $b$  by deleting several (possibly, zero or all) elements.

### Input

The first line contains  $T$  ( $1 \leq T \leq 100$ ), the number of independent test cases.

Each of the next lines contains two integers  $A$  and  $B$  ( $1 \leq A, B \leq 10^{18}$ ).

### Output

Output the number of different subsequences of  $gen\_string(A, B)$  modulo 998244353.

## Examples

standard input	standard output
6	4
1 1	70
3 5	264
4 7	196417
8 20	609
4 10	667131122
27 21	
18	988
5 9	220693002
23 30	133474535
820 483	202371605
5739 9232	778839228
86494 55350	282057418
606 13336	935955056
2768848 1124639	943144752
47995594 66053082	409056617
1069395 7177	627433544
7801842511 4390103762	578769776
47882886553 82678306054	917438628
193410894 6189355686	24364208
51594638 19992922190	109943645
59 110	352575425
422735115778072 658356435030265	68058533
9125338158530266 5328357177709583	402004723
60743352262021049 95595862538791630	894026897
629312141725417942 999581828389011547	

## Problem C. Palindrome

Input file:            standard input  
Output file:           standard output  
Time limit:           5 seconds  
Memory limit:        1024 megabytes

As a magician and a palindrome lover, you want to make strings become palindromes through magic operation.

In one magic operation, you can erase  $S[l...r]$  of a string  $S$  and concatenate the rest of  $S$  to get the target string, which costs  $r - l + 1$  units of magic potion.

You are given a string  $str$ , consisting of  $n$  lowercase Latin letters, and there are  $m$  magic tests.

For each one, you are given two integers  $l, r$ , denoting  $S$  as  $str[l...r]$ .

You should use **at most one** magic operation, report the minimal cost of magic potion to make  $S$  become palindrome, and the number of ways to achieve the target with the previous minimized cost.

Specifically, if  $S$  is already a palindrome, just output '0 0'.

### NOTE:

- A palindrome is a string that reads the same from left to right as from right to left. For example, 'aba', 'ccpcc', 'qaq' are palindromes, while 'ccpc' and 'qhd' are not.
- $S[l...r]$  means the substring of  $S$  which starts from the  $l$ -th character and ends with the  $r$ -th character.

### Input

The first line contains an integer  $n$  and a string  $str$  ( $1 \leq n = |str| \leq 5 \times 10^5$ ) of lowercase English letters.

The second line contains an integer  $m$  ( $1 \leq m \leq 4 \times 10^5$ ) representing the number of magic tests.

The following  $m$  lines describe the tests.

In each line, there are two integers  $l, r$  ( $1 \leq l \leq r \leq n$ ), you should take the  $str[l...r]$  as the problem.

### Output

For each tests, output one line consisting two integers - the minimal cost and the number of ways to achieve it, separated by one space.

### Examples

standard input	standard output
5 abcca	1 1
3	0 0
1 5	1 1
3 4	
3 5	
5 babdb	1 1
2	1 2
1 4	
3 4	

## Problem D. Yet Another Coffee

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         256 megabytes

The girls of SYSU like drinking tea. But one day, they wanted a change and decided to try coffee in the next  $n$  days.

Now Zayin, who always provides food and drinks for SYSU, will go to the shop to buy some coffee. She learns that the price of day  $i$  is  $a_i$ . Meanwhile, she has  $m$  coupons - the  $i$  coupon can be used before day  $r_i$  (inclusively) and can reduce the price of coffee on that day by  $w_i$ .

**Notice that each coupon can be used only once and Zayin can use more than one coupon per day. The price can be a negative number after using the coupons.**

Since the girls of SYSU still need to drink tea, Zayin decided to choose some days to buy coffee. Now she wants to know the minimum money she will spend (or the maximum money she can get) if she chooses exactly  $k$  ( $1 \leq k \leq n$ ) days to buy coffee.

### Input

The first line of the input contains a single integer  $t$  ( $1 \leq t \leq 10$ ) — the number of test cases. The description of the test cases follows.

The first line contains two integers  $n, m$  ( $1 \leq n, m \leq 2 \times 10^5$ ) - the number of days and the number of coupons.

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) -  $a_i$  denotes the price of coffee on day  $i$ .

Next  $m$  lines, each line contains two integers  $r_i, w_i$  ( $1 \leq r_i \leq n, 1 \leq w_i \leq 10^9$ ), denoting that the  $i$  th coupon can be used before day  $r_i$  and can reduce the price by  $w_i$ .

### Output

For each testcase, output  $n$  integers - the  $i$  th integer represents the minimum money Zayin will spend to buy coffee if she chooses exactly  $i$  days to buy coffee.

**Notice that the answer can be a negative integer.**

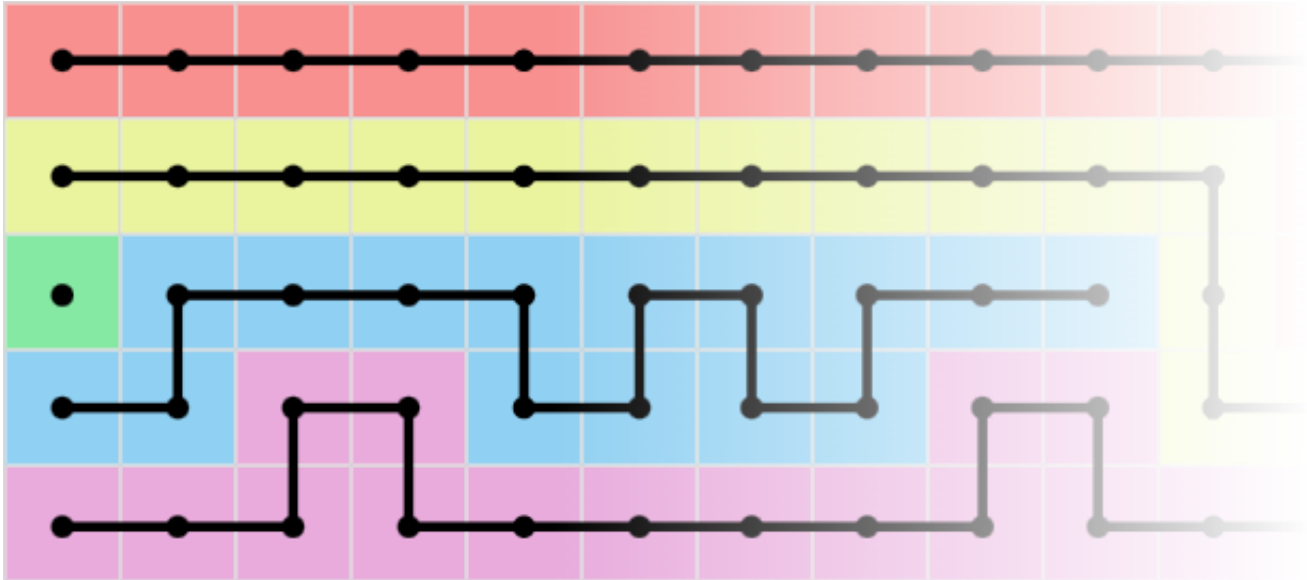
### Example

standard input	standard output
2	-2 0 3 7 12
5 2	-21 -18 -15 -11 -5 3 13
1 2 3 4 5	
3 1	
4 2	
7 3	
4 3 1 10 3 8 6	
4 9	
3 8	
4 5	

## Problem E. Coloring Tape

Input file:            standard input  
 Output file:         standard output  
 Time limit:          1.5 seconds  
 Memory limit:       256 megabytes

Imagine you have a large tape of size  $n \times m$ , with  $n$  brushes dipped in different colors arranged in the first column, waiting to color the entire tape.



This image shows an example, where the lines represent the movement path of the brushes.

For each brush, you can move it to the right, up or down for several times. Each cell **cannot be colored twice**, even if the colors are the same. Your goal is simple - to color **all cells** of the tape using some operations. Note that the initial positions of the brushes are already colored.

Meanwhile, due to the decorative purpose of the tape, there are some restrictions on coloring. For each restriction, we will provide **two cells in the same column** and indicate whether the colors of these two cells must be the same or different.

Under these restrictions, how many different color arrangements can the final tape have? Note that in this case, we do not consider flipping or rotating the tape.

### Input

The first line contains three integers:  $n$  ( $1 \leq n \leq 14$ ),  $m$  ( $2 \leq m \leq 500$ ), and  $r$  ( $0 \leq r \leq 500$ ), representing the width, length, and number of restrictions of the tape, respectively.

Next, there are  $r$  lines, each containing four integers:  $c \ x \ y \ diff$  ( $1 \leq c \leq m, 1 \leq x < y \leq n, diff \in \{0, 1\}$ ). Here,  $diff = 1$  means that the colors of the  $x$ -th and  $y$ -th cells in column  $c$  must be different, otherwise they must be the same.

### Output

Output the result modulo 998244353.

## Examples

standard input	standard output
3 5 3 3 2 3 0 4 1 2 0 5 1 3 0	19
5 10 10 9 3 4 1 2 4 5 0 7 2 3 0 9 2 3 0 6 3 5 0 6 2 4 1 2 4 5 0 1 1 3 1 7 2 4 0 10 2 3 0	1514
4 2 0	17

## Problem F. Mystery of Prime

Input file:            **standard input**  
 Output file:          **standard output**  
 Time limit:            **1 second**  
 Memory limit:         **256 megabytes**

The prime minister of the Mathematical Kingdom is a crazy follower of prime numbers, therefore he especially set up a department to solve problems related to prime numbers. The head of this department, Mr.Robert, recently received a letter from his friend Euler.

The letter contains a mystery of prime. There is a sequence  $a_1, a_2, \dots, a_n$ . Euler considers a sequence beautiful if and only if all elements are **positive** integers and the sum of any two **adjacent** elements is a prime.

Formally,  $\forall i \in [1, n] \cap \mathbb{N}, a_i \in \mathbb{N}^+, \forall i \in [1, n) \cap \mathbb{N}, (a_i + a_{i+1}) \in \mathbb{P}$ , where  $\mathbb{P}$  represents the set containing all primes.

Sometimes, the given sequence is not beautiful. Mr.Robert would like to make the least effort to make it beautiful, that is to modify the elements of the sequence and minimize the number of updated elements.

Mr.Robert is busy these days, so he asked you to report the minimum number of updated elements to make the sequence beautiful.

### Input

The first line contains an integer  $n$  ( $2 \leq n \leq 10^5$ ).

The second line contains  $n$  positive integers, representing  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^5$ ).

### Output

Print a single integer representing the minimum number of updated elements to make the sequence beautiful.

### Examples

standard input	standard output
6 1 5 1 4 4 1	2
9 30 6 7 12 15 8 20 17 14	4

### Note

For the first test, the updated sequence may be "1 2 1 1 4 1".



## Problem G. Path

Input file:            **standard input**  
 Output file:          **standard output**  
 Time limit:           1 second  
 Memory limit:        256 megabytes

Given an array  $a$  of length  $n$  and an array  $b$  of length  $m$ , construct a grid of size  $n \times m$ , where the value in cell  $(x, y)$  is denoted as  $C[x, y]$  and calculated as  $a_x + b_y$ .

You start from  $(1, 1)$ , and in each step, you choose a grid cell located at the bottom right to move to, until you reach  $(n, m)$ , aiming to maximize the sum of absolute differences between adjacent cells along the path.

Formally, your goal is to find a sequence  $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$  that satisfies the conditions

- $(x_1, y_1) = (1, 1)$
- $(x_k, y_k) = (n, m)$
- $x_i \leq x_{i+1}, y_i \leq y_{i+1}, (x_i, y_i) \neq (x_{i+1}, y_{i+1}) \forall i \in [1, k]$

while maximizing the  $\sum_{i=1}^{k-1} |C[x_i, y_i] - C[x_{i+1}, y_{i+1}]|$ .

### Input

The first line contains two integers,  $n, m$  ( $1 \leq n, m \leq 10^5$ ).

The second line contains  $n$  integers, representing the array  $a$  ( $1 \leq a_i \leq 10^5$ ).

The third line contains  $m$  integers, representing the array  $b$  ( $1 \leq b_i \leq 10^5$ ).

### Output

One line with an integer representing the answer.

### Examples

standard input	standard output
4 4 1 3 3 1 8 10 8 5	11
4 2 5 7 8 10 10 3	12

## Problem H. Quake and Rebuild

Input file:            standard input  
Output file:           standard output  
Time limit:            3 seconds  
Memory limit:         256 megabytes

The Kingdom of Calamita was a small nation constantly plagued by natural disasters. The people built their villages and towns in the shades of the great Banyan tree that stood at the kingdom's center. Its massive roots provided stability to the land and the villagers always looked to it for guidance and protection.

The great tree is a rooted tree whose root is node 1. For other node  $i$  ( $i > 1$ ), its parent in the tree is  $fa[i]$  ( $1 \leq fa[i] < i$ ) and there is an edge in between. There were two types of events that took place in the Kingdom of Calamita — **quake and rebuild**.

1. **Quake**: When a tremor hit the range  $[l, r]$  with a magnitude of  $d$ , it caused the parent node of every node in that range to shift down by  $d$ . In other words,  $\forall i \in [l, r], fa'[i] = \max(1, fa[i] - d)$ . The tree structure shifts after that.

2. **Rebuild**: During the rebuilding after some disasters, the villagers were seeking for economic recovery. To boost the economy, they set a list of nodes  $[b_1, b_2, \dots, b_k]$  to become the new trade centers and decided to build some temporary transportation hubs to connect all of the new trade centers. Two trade centers are connected if and only if every node on the simple path between these two centers has been built with a temporary transportation hub on it. Obviously, the villagers were only interested in the minimal number of temporary transportation hubs to achieve the goal.

Over time there were  $m$  such events — some quakes shook up the root system, and some rebuilding aimed at economic recovery.

For each rebuild plan, you need to help the villagers figure out the minimal number of temporary transportation hubs to make all trade centers connected.

Note that Two rebuild events are **independent**, i.e., the temporary transportation hubs will shut down after the previous rebuilding event, and if you want to use it again, you need to rebuild it.

### Input

The first line contains two integers  $n, m$  ( $2 \leq n, m \leq 2 \times 10^5$ ) respectively representing the number of nodes in the tree and the number of events.

The second line contains an integer  $n - 1$  integers, the  $i$ -th integer  $fa[i + 1]$  denotes the parent of node  $i + 1$ .

The following  $m$  lines describe the events.

For the quake, you are given four positive integers  $1, l, r, d$  ( $2 \leq l \leq r \leq n, d \leq n$ ).

For the rebuilding, you are firstly given two positive integers 2 and  $k$ , then following  $k$  positive integers representing  $b_1, b_2, \dots, b_k$ . The list of node's indexes may coincide, you can just ignore the duplication and keep only one of the same indexes.

It's guaranteed that  $\sum k \leq 6 \times 10^5$  across all rebuilding events.

### Output

For each rebuild event, you should output one line consisting one positive integer representing the minimal number of the temporary transportation hubs to make the trade centers connected.

## Examples

standard input	standard output
4 5 1 2 2 2 2 1 4 1 2 3 1 2 3 2 3 4 1 4 4 1 2 2 3 4	3 4 3
10 10 1 2 3 3 4 5 7 7 9 1 2 10 3 2 9 9 5 3 10 7 2 4 6 8 1 6 10 3 1 2 7 3 1 7 10 3 2 2 4 3 2 3 7 4 4 1 3 9 3 1 3 9 3 1 10 10 3	10 3 3

## Problem I. Phony

Input file:            **standard input**  
 Output file:          **standard output**  
 Time limit:           **2 seconds**  
 Memory limit:        **256 megabytes**

You are given a sequence of length  $n$  and the number  $k$ , ask for support  $m$  operations or inquiries:

- C  $t$ : Let the largest number minus  $k$  (the far left one if there is more than one), executed  $t$  times.
- A  $x$ : Ask for the  $x$ -th largest number.

### Input

The first line contains three integer  $n$ ,  $m$  and  $k$  ( $1 \leq n, m \leq 5 \times 10^5$ ).

The next line contains  $n$  integers representing the sequence  $a$  ( $1 \leq a_i \leq 10^{18}$ ).

Each of the following  $m$  lines contains a character and an integer, indicating an operation or inquiry.

For all data guaranteed  $1 \leq k, t \leq 10^{18}$ ,  $1 \leq x \leq n$ .

Ensure that the sequence after each operation for all  $a_i$  satisfies  $-10^{18} \leq a_i \leq 10^{18}$

### Output

For each inquiry, print a line for the answer.

### Example

standard input	standard output
3 5 5	3
7 3 9	4
A 3	-1
C 1	
A 2	
C 2	
A 3	

## Problem J. Keyi Likes Reading

Input file:            standard input  
 Output file:          standard output  
 Time limit:           1 second  
 Memory limit:        256 megabytes

Keyi has a strong passion for reading, and she has developed a habit of morning reading every day.

With the college entrance exam approaching, Keyi plans to learn several words from the 'Victor Dictionary' each morning.

However, her method of memorizing words is a bit peculiar. If she learns a word of length  $k$  today, she insists on memorizing **all** the words in the dictionary that have the same length of  $k$ .

But Keyi's daily energy is limited, and she cannot learn more than  $W$  words in a day, or else she won't be able to remember them all.

Keyi is unsure how to efficiently plan which words to study each morning, so that it can minimize the number of days it will take her to complete the entire 'Victor Dictionary'. Therefore, she kindly requests your assistance.

### Input

The first line contains two integers,  $n$  and  $W$  ( $1 \leq W \leq n \leq 50000$ ).

The second line contains  $n$  integers  $a_i$  ( $1 \leq a_i \leq 13$ ) -  $a_i$  denotes the length of the  $i$  ( $1 \leq i \leq n$ )-th word.

It's guaranteed that for all words of the same length, they will not occur more than  $W$  times.

### Output

Output the minimum number of days it takes to learn the entire "Victor Dictionary."

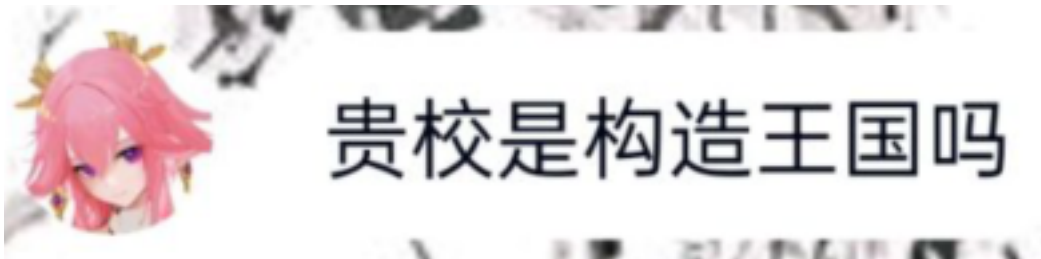
### Example

standard input	standard output
5 4 1 2 1 2 1	2

## Problem K. Make SYSU Great Again II

Input file:            standard input  
Output file:           standard output  
Time limit:            2 seconds  
Memory limit:         256 megabytes

As you know, you guys are participating in CCPC (Chinese Constructive Problem Contest), and no doubt that the proposition school SYSU, as the kingdom of constructive problems, is going to challenge you to solve some related problems.



Given an  $n \times n$  grid, you are asked to fill **each cell** with a number within  $[0, 4n^2 - 1]$ , and it needs to fulfill the following requirements.

1. Each number occurs at most 5 times.
2. Each cell should be filled with exactly one number.
3. For any two adjacent cells with a common edge, their bitwise AND value should be exactly equal to 0.

Construct a valid solution, or report it's impossible.

### Input

The only line contains a single number  $n$  ( $1 \leq n \leq 2000$ ), representing the size of the grid.

### Output

If a valid solution exists, first output *Yes* on a single line. Then, output an  $n \times n$  integer matrix, where each number is in the range  $[0, 4n^2 - 1]$ , representing the constructed matrix.

If there are multiple valid solutions, output any one of them.

If a valid solution does not exist, just output *No* on a single line.

### Example

standard input	standard output
4	Yes 0 0 0 0 0 1 2 1 1 2 1 2 2 1 2 4

## Problem L. Yet Another Maximize Permutation Subarrays

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1.5 seconds  
Memory limit:         256 megabytes

You are given a permutation  $p$  of size  $n$ . You want to maximize the number of subarrays of  $p$  that are permutations. In order to do so, you must perform the following operation exactly once:

- Select integers  $i, j$ , where  $1 \leq i, j \leq n$ , then
- Swap  $p_i$  and  $p_j$ .

For example, if  $p = [5, 1, 4, 2, 3]$  and we choose  $i = 2, j = 3$ , the resulting array will be  $[5, 4, 1, 2, 3]$ . If instead we choose  $i = j = 5$ , the resulting array will be  $[5, 1, 4, 2, 3]$ .

Which choice of  $i$  and  $j$  will maximize the number of subarrays that are permutations?

### NOTE:

- A permutation of length  $n$  is an array of  $n$  distinct integers from 1 to  $n$  in arbitrary order. For example,  $[2, 3, 1, 5, 4]$  is a permutation, but  $[1, 2, 2]$  is not a permutation (2 appears twice in the array), and  $[1, 3, 4]$  is also not a permutation ( $n = 3$  but there is 4 in the array).
- An array  $a$  is a subarray of an array  $b$  if  $a$  can be obtained from  $b$  by deleting several (possibly, zero or all) elements from the beginning and several (possibly, zero or all) elements from the end.

### Input

The first line of the input contains a single integer  $t$  ( $1 \leq t \leq 10$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer  $n$  ( $1 \leq n \leq 10^6$ ) — the size of the permutation.

The next line of each test case contains  $n$  integers  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq n$ , all  $p_i$  are distinct) — the elements of the permutation  $p$ .

### Output

For each test case, output two integers  $i$  and  $j$  ( $1 \leq i, j \leq n$ ) — the indices to swap in  $p$ .

If there are multiple solutions, print any of them.

## Example

standard input	standard output
8	3 3
3	1 2
1 2 3	1 4
3	1 3
1 3 2	9 9
5	4 9
1 3 2 5 4	2 4
6	1 5
4 5 6 1 2 3	
9	
8 7 6 3 2 1 4 5 9	
10	
7 10 5 1 9 8 3 2 6 4	
10	
8 5 10 9 2 1 3 4 6 7	
10	
2 3 5 7 10 1 8 6 4 9	



## Problem M. Inverted

Input file:            **standard input**  
 Output file:          **standard output**  
 Time limit:           **1 second**  
 Memory limit:        **256 megabytes**

Given a tree with  $n$  nodes initially numbered from 1 to  $n$ , and a node sequence of  $n - 1$  length, we are going to perform operations on the tree according to the order of the sequence.

For each operation, if the node to be operated is  $x$ , firstly create a new node numbered  $x + n$ . For any integer  $i \in [1, n]$  that the edge  $(x, i)$  exists:

- If the node  $i + n$  does not exist, we connect  $(x + n, i)$ .
- If the node  $i + n$  exists (in this case, the edge  $(x, i + n)$  always exists), we connect  $(x + n, i + n)$  and delete edge  $(x, i + n)$ .

For the resulting graph after each operation, calculate the number of spanning trees modulo 998244353.

### Input

The first line contains an integer  $n$  ( $1 \leq n \leq 5000$ ), indicating the size of the tree.

The next  $n - 1$  lines each contain two numbers  $u$  and  $v$  ( $1 \leq u, v \leq n$ ), representing an edge  $(u, v)$  in the tree. It is guaranteed that the input forms a valid tree.

The next line contains  $n - 1$  **distinct** numbers  $b_i$  ( $1 \leq b_i \leq n$ ), representing the sequence of nodes to be operated in order.

### Output

Output  $n - 1$  lines, the only number in  $i$ -th line represents the number of spanning trees in the graph after the  $i$ -th operation, modulo 998244353.

### Example

standard input	standard output
5	4
1 2	4
1 3	6
2 4	1
2 5	
1 5 2 3	