

# The 3rd Universal Cup



## Stage 3: Ukraine

June 29-30, 2024

This problem set should contain 12 problems on 18 numbered pages.

Based on



Osijek Competitive Programming Camp

## Problem A. Aibohphobia

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

*It's not a situation, it's a condition. Electromagnetic Hypersensitivity. For reasons unknown, my nervous system has become sensitized to certain frequencies of electromagnetic radiation.*

—Chuck McGill, *Better Call Saul*

Chuck McGill suffers from electromagnetic hypersensitivity: he is afraid of electricity. But did you know he is also afraid of palindromes?

Once he received a string  $s$  as a birthday present from Jimmy. He wants to rearrange its symbols, obtaining string  $t$ , so that the following condition holds:

- For every  $i = 2, 3, \dots, |s|$ , string  $t_1t_2 \dots t_i$  is **not** a palindrome.

Can you help him?

As a reminder, a string  $s$  is called a **palindrome**, if it reads the same backwards as forwards. For example, `aibohphobia` is a palindrome.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^5$ ) — the number of test cases. The description of test cases follows.

The only line of each test case contains a string  $s$ , consisting of lowercase Latin letters.

It is guaranteed that the sum of lengths of  $s$  over all test cases does not exceed  $10^6$ .

### Output

For each test case, if there is no such way to rearrange characters of  $s$ , print `N0`.

Otherwise, print `YES`. On the next line output a single string  $t$ . It should be the rearrangement of characters of  $s$ , and it should satisfy the condition from the statement.

You can print `YES` and `N0` in any case (e.g. the strings `yEs`, `yes`, `Yes` will be taken as a positive answer).

## Example

standard input	standard output
5	YES
a	a
sos	YES
abba	oss
icpc	NO
tenet	YES
	icpc
	YES
	tente

## Problem B. Breaking Bad

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

*Nah, come on, man. Some  
straight like you, \*\*\*\*\* \*\*  
\*\*\* \*\* all a sudden at age,  
what, 60, he's just gonna break  
bad?*

—Jesse, *Breaking Bad*

Walter White finally broke bad and finished building his drug empire.

There are 5 of them in the business right now: Walt, Jesse, Mike, Saul, and Todd. They divided Albuquerque into  $n \times n$  cells, and they made  $a_{i,j}$  dollars in the area on the intersection of the  $i$ -th row and  $j$ -th column.

They want to choose  $n$  cells to collect money from. To not cause Hank's suspicion, they want exactly one selected cell in each row and in each column. If the total money made in these cells is  $S$ , they will split it evenly among the five of them, and will donate the remaining  $S \bmod 5$  dollars to Ted Beneke.

Find all possible amounts of dollars that could have been donated to Ted Beneke.

### Input

The first line contains a single positive integer  $n$  ( $1 \leq n \leq 10^3$ ) — the size of the grid.

The next  $n$  lines contain  $n$  integers  $a_{i,j}$  ( $0 \leq a_{i,j} \leq 4$ ) — the number of dollars made in the cell at the intersection of the  $i$ -th row and  $j$ -th column.

### Output

Output a string of exactly 5 letters Y or N. If it's possible to get  $S \equiv i \pmod 5$  for ( $0 \leq i < 5$ ), the  $i$ -th letter of the string must be Y, otherwise N.

### Examples

standard input	standard output
2 0 4 4 0	YNNYN
2 1 1 1 1	NNYNN

## Problem C. Chemistry Class

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

*I am an extremely overqualified  
high school chemistry teacher.  
When I can work, I make  
\$43,700 per year.*

—Walter White, *Breaking Bad*

Walter White has  $2n$  students in his chemistry class. Student  $i$  has chemistry skill  $a_i$ .

He wants to divide students into  $n$  pairs for a group exercise. The pair works better together if their skills are closer. More precisely,

- If skills of students in a pair differ by more than  $A$ , they will **blow up the lab**;
- If skills of students in a pair differ by at most  $A$ , but by more than  $B$ , they will **produce a mediocre product**;
- If skills of students differ by at most  $B$ , they will **produce 99.1% pure product**.

Walter wants to split students into  $n$  pairs so that:

- Lab is not blown up;
- The number of pairs that produced 99.1% pure product is as large as possible.

Determine if Walter can split students in such a way, and if he can, find the largest possible number of pairs that would produce 99.1% pure product.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^5$ ) — the number of test cases. The description of test cases follows.

The first line of each test case contains three integers  $n, A, B$  ( $1 \leq n \leq 2 \cdot 10^5, 1 \leq B < A \leq 10^{18}$ ) — the number of students.

The second line of each test case contains  $2n$  integers  $a_1, a_2, \dots, a_{2n}$  ( $0 \leq a_i \leq 10^{18}$ ) — skills of the students.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $2 \cdot 10^5$ .

### Output

For each test case, if there is no way to split the students into pairs without blowing up the lab, output  $-1$ . Otherwise, output the largest possible number of pairs that would produce 99.1% pure product.

## Example

standard input	standard output
4	-1
1 2 1	2
42 69	1
2 3 1	4
1 2 3 4	
2 5 1	
6 1 3 4	
5 19 1	
1 7 8 9 10 11 12 13 14 20	

## Note

In the first test case, it's impossible to split students into pairs without blowing up the lab.

In the second test case, we can pair the first student with the second, and the third one with the fourth one. Both pairs will have a difference in skills equal to 1, and both will produce 99.1% pure product.

## Problem D. Daily Disinfection

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

*We're gonna scour every vat,  
every keg, every cook surface.  
We've gotta clean up every  
possible source of contamination,  
and only then we cook.  
Comprende?*

—Jesse, *Breaking Bad*

To get 99.1% pure product, everything in the lab has to be thoroughly cleaned daily. Right now, Jesse is going to clean a shelf with books.

The shelf consists of  $n$  places for books. There are some books placed on the shelf (possibly, none). If some place is empty, Jesse can clean it without any power consumption (he is really good at cleaning). He cannot clean any place with a book currently in it. If there is a book at some place, and the adjacent place is empty, Jesse can move the book to that adjacent place consuming 1 power.

Jesse is a very busy person, and he does not want to spend too much power. For each test case, tell the minimal amount of power Jesse has to spend.

### Input

The first line contains of the single integer  $t$  ( $1 \leq t \leq 10^5$ ) — the number of the test cases.

The first line of each of the  $t$  test cases contains a single integer  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — the number of the places on the shelf.

The second line contains a string  $s$  of length  $n$ , where  $s_i = 0$  if there is no book at  $i$ -th place, and  $s_i = 1$  otherwise.

It is guaranteed that in all provided test cases it is possible to clean the shelf.

The sum of  $n$  over all test cases will not be greater than  $2 \cdot 10^5$ .

### Output

Output  $t$  lines, each line containing a single integer — the minimal amount of power Jesse has to spend.

### Example

standard input	standard output
3	1
2	2
01	6
5	
00110	
9	
101010101	

### Note

In the first test case, Jesse can clean the first place, then move the book from the second place to the first (consuming 1 power), and then clean the second place.

## Problem E. Equalizer Ehrmantraut

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

*You know how they say, 'It's  
been a pleasure?' It hasn't.*

—Mike Ehrmantraut, *Breaking  
Bad*

When Mike Ehrmantraut started being a cop, he wanted all people to be equal before the law. Now he is retired, but he still prefers **almost equal** arrays.

Here two arrays  $a$  and  $b$  of length  $n$  are called **almost equal** if the following condition holds:

- For any  $1 \leq i < j \leq n$ ,  $\min(a_i, b_j) = \min(a_j, b_i)$

Given two integers  $n$  and  $m$ , find the number of **almost equal** pairs  $(a, b)$  of integer arrays of length  $n$  with elements from 1 to  $m$ . As this number can be large, output it modulo 998244353.

### Input

The only line of the input contains two integers  $n, m$  ( $1 \leq n, m \leq 10^6$ ).

### Output

Output a single integer — output to the problem modulo 998244353.

### Examples

standard input	standard output
1 3	9
2 2	10
69 42	608932821

### Note

In the first sample, any pair of arrays  $([x], [y])$  with  $1 \leq x, y \leq 3$  satisfies the conditions from the statement, there are 9 of them.

In the second sample, there are the 10 pairs of arrays:  $([1, 1], [1, 1])$ ,  $([1, 1], [1, 2])$ ,  $([1, 1], [2, 1])$ ,  $([1, 1], [2, 2])$ ,  $([1, 2], [1, 1])$ ,  $([1, 2], [1, 2])$ ,  $([2, 1], [1, 1])$ ,  $([2, 1], [2, 1])$ ,  $([2, 2], [1, 1])$ ,  $([2, 2], [2, 2])$ .



## Problem F. Formal Fring

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

*I don't think we're alike at all,  
Mr. White. You're not a  
cautious man at all... [Y]ou have  
poor judgement. I cannot work  
with someone with poor  
judgement.*

—Gustavo Fring, *Breaking Bad*

Gus Fring is a very careful man. He doesn't mess with you. When he asks someone to solve a problem, he gives a completely formal statement. And today he asked YOU.

For a positive integer  $n$ , define  $highest\_bit(n)$  as the largest number  $i$ , such that  $2^i \leq n$ . Also define  $highest\_bit(0) = -1$ .

You are given a positive integer  $X$ . Find the number of multisets  $S$  of positive integers, which satisfy the following conditions:

- All elements of  $S$  are nonnegative powers of 2.
- The sum of elements of  $S$  is  $X$ .
- There is no way to split elements of  $S$  into two groups so that  $highest\_bit(S_1) = highest\_bit(S_2)$ , where  $S_1$  is the sum of the elements in the first group, and  $S_2$  is the sum of the elements in the second group.

Solve this problem for  $X = 1, 2, \dots, n$ .

Since the answers can be very large, output them modulo 998244353.

### Input

The only line of the input contains a single integer  $n$  ( $1 \leq n \leq 10^6$ ).

### Output

Output  $n$  integers: answers to the problem for  $X = 1, 2, \dots, n$ , modulo 998244353.

### Example

standard input	standard output
10	1 1 2 1 1 3 6 1 1 2

## Problem G. Goodman

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

*S'all good, man.*

—James McGill, *Better Call Saul*

Saul Goodman is the best lawyer in the world. He believes that until proven guilty, every man, woman, and child in this country is innocent. However, one client of Saul is not so innocent.

Saul needs to build a solid story for his defense. There were  $n$  events, numbered from 1 to  $n$ . Saul needs to come up with an order  $q_1, q_2, \dots, q_n$ , in which these events happened. His client, however, has a poor memory! If Saul tells him to remember permutation  $(q_1, q_2, \dots, q_n)$  of events, he will remember permutation  $(p_{q_1}, p_{q_2}, \dots, p_{q_n})$ , where  $(p_1, p_2, \dots, p_n)$  is some (fixed) permutation of integers from 1 to  $n$ .

Saul wants their permutations to be as consistent as possible: he wants to maximize the value of  $LCS((q_1, q_2, \dots, q_n), (p_{q_1}, p_{q_2}, \dots, p_{q_n}))$ . Help him to find permutation  $(q_1, q_2, \dots, q_n)$  which maximizes this value! If there are many such permutations, find any of them.

Here  $LCS(a, b)$  denotes the length of the longest common subsequence of sequences  $a$  and  $b$ . For example,  $LCS((1, 3, 4, 2, 5), (3, 1, 2, 4, 5)) = 3$ , and one of the common subsequences of length 3 is  $(1, 2, 5)$ .

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^5$ ) — the number of test cases. The description of test cases follows.

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

The second 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.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $10^6$ .

### Output

For each test case, output  $n$  integers  $q_1, q_2, \dots, q_n$  ( $1 \leq q_i \leq n$ , all  $q_i$  are distinct) — any permutation of integers from 1 to  $n$ , which **maximizes** the value of  $LCS((q_1, q_2, \dots, q_n), (p_{q_1}, p_{q_2}, \dots, p_{q_n}))$ .

### Example

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

### Note

In the first test case, for  $q = (1, 2, 3, 4)$ , we have  $LCS((1, 2, 3, 4), (p_1, p_2, p_3, p_4)) = LCS((1, 2, 3, 4), (1, 2, 3, 4)) = 4$ .

In the second test case, for  $q = (1, 6, 2, 5, 3, 4)$ , we have:

$LCS((1, 6, 2, 5, 3, 4), (p_1, p_6, p_2, p_5, p_3, p_4)) = LCS((1, 6, 2, 5, 3, 4), (6, 1, 5, 2, 4, 3)) = 3$ ; one of common subsequences of length 3 is  $(1, 2, 3)$ .

## Problem H. Highway Hoax

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

*You think this is something?  
You think this is bad? This?  
This chicanery?*

---

—Chuck McGill, *Better Call Saul*

Albuquerque is a large city. It has  $n$  crossings and  $n - 1$  one-directional highways between them. It's guaranteed that if all highways were undirected, it would be possible to get from any crossing to any other by these highways. Additionally, every crossing has a label: **S** or **F**, indicating start and finish correspondingly.

Saul and Kim are going to do another chicanery. They can perform the following operation any number of times:

- Choose any highway  $(u, v)$  such that the following conditions hold:
  - This highway is directed from crossing  $u$  to crossing  $v$
  - Crossing  $u$  has label **S**
  - Crossing  $v$  has label **F**
- Change the direction of the highway and swap labels on  $u$  and  $v$ .

What is the total number of different configurations that Saul and Kim can achieve by applying this operation any number of times? As this number can be very large, output it modulo 998244353.

Two configurations are called different if some crossing has different labels in them, or if some highway has different orientation in them.

### Input

The first line of the input contains a single integer  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) — the number of crossings in Albuquerque.

The second line of the input contains a string of length  $n$ , consisting of characters **S** and **F**. The  $i$ -th character of this string denotes the initial label of crossing  $i$ .

The  $i$ -th of the next  $n - 1$  lines contains two integers  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i$ ) — denoting a directed highway from crossing  $u_i$  to crossing  $v_i$ . It's guaranteed that if all highways were undirected, it would be possible to get from any crossing to any other by these highways.

### Output

Output a single integer — the total number of different configurations that Saul and Kim can achieve by applying this operation any number of times, modulo 998244353.

## Examples

standard input	standard output
5 SFSFS 2 1 2 3 4 3 4 5	1
4 SFFF 1 2 1 3 1 4	4
7 SSSSFFF 1 2 3 2 4 3 4 5 4 6 2 7	13

## Note

In the first sample, all highways are directed from crossings with **F** on them to crossings with **S** on them, so it's impossible to do any operations.

In the second sample, for each  $v = 2, 3, 4$  it's possible to do an operation with nodes  $1, v$ . Resulting three configurations, together with the initial one, are the only possible configurations.

## Problem I. Increasing Income

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

*Everything i did, I did it for me.*

—Walter White, *Breaking Bad*

Walter always says he is doing everything for his family. But in reality, due to his ego, he became obsessed with making as much money as possible.

It turned out that his income would be equal to the maximal possible value of  $f((q_1, q_2, \dots, q_n)) + f((p_{q_1}, p_{q_2}, \dots, p_{q_n}))$  over all permutations  $(q_1, q_2, \dots, q_n)$  of integers from 1 to  $n$ .

Here  $p_1, p_2, \dots, p_n$  is Walt's favorite permutation, and  $f(a)$  is defined as the number of positions  $1 \leq i \leq n$ , for which  $a_i = \max(a_1, a_2, \dots, a_i)$ : in other words, the number of prefix maximums.

Find any permutation  $(q_1, q_2, \dots, q_n)$  of integers from 1 to  $n$  that maximizes Walt's income. If there are many such permutations, find any of them.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^5$ ) — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — the length of Walt's favorite permutation.

The second 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) — elements of the permutation.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $2 \cdot 10^5$ .

### Output

For each test case, output  $n$  integers  $q_1, q_2, \dots, q_n$  ( $1 \leq q_i \leq n$ , all  $q_i$  are distinct) — any permutation of integers from 1 to  $n$ , which **maximizes** the value of  $f((q_1, q_2, \dots, q_n)) + f((p_{q_1}, p_{q_2}, \dots, p_{q_n}))$

### Example

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

### Note

In the first test case, for  $q = (1, 2, 3)$ , the value is  $f(1, 2, 3) + f(p_1, p_2, p_3) = f(1, 2, 3) + f(1, 2, 3) = 3 + 3 = 6$ .

In the second test case, for  $q = (1, 3, 4, 2)$ , the value is  $f(1, 3, 4, 2) + f(p_1, p_3, p_4, p_2) = f(1, 3, 4, 2) + f(2, 3, 1, 4) = 3 + 3 = 6$ .

In the third test case, for  $q = (1, 3, 5, 4, 2)$ , the value is  $f(1, 3, 5, 4, 2) + f(p_1, p_3, p_5, p_4, p_2) = f(1, 3, 5, 4, 2) + f(1, 2, 3, 4, 5) = 3 + 5 = 8$ .

## Problem J. Jesse's Job

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

*You had one thing to do, one thing! That is the only thing, I might add, that might have saved our lives.*

—Walter White, *Breaking Bad*

Jesse has a permutation  $p_1, p_2, \dots, p_n$  of integers from 1 to  $n$ . His job is simple: to maximize the number of positions  $i$ , for which  $p_i = i$ . To achieve that, Jesse can do the following operation **exactly once**:

- Color some elements of the permutation in yellow, and all the remaining elements in blue. There has to be at least one yellow and at least one blue element.
- Then, separately sort yellow and blue numbers.

For example, for permutation  $(3, 5, 1, 6, 2, 4)$ , Jesse can mark numbers 3, 5, 4 yellow, and 1, 6, 2 blue. After sorting yellow and blue elements separately, Jesse will get permutation  $(3, 4, 1, 2, 6, 5)$ .

Jesse's score in the end is the number of positions  $i$ , for which  $p_i = i$ . Find the maximal score Jesse can achieve and some way to achieve it.

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^5$ ) — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer  $n$  ( $2 \leq n \leq 10^6$ ) — the length of the permutation.

The second 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) — elements of the permutation.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $10^6$ .

### Output

For every test case, in the first line, output the maximal score Jesse can achieve.

In the second line, output a single integer  $k$  ( $1 \leq k \leq n - 1$ ) — the number of elements Jesse should color yellow.

In the third line, output  $k$  integers  $pos_1, pos_2, \dots, pos_k$  ( $1 \leq pos_i \leq n$ , all  $pos_i$  are distinct) — the **positions** of elements that you are going to color in yellow.

If there are several ways to obtain the maximum score, output any of them.

## Example

standard input	standard output
3	0
2	1
2 1	1
4	4
2 1 4 3	2
6	1 2
3 5 4 2 6 1	4
	3
	1 3 4

## Note

In the first test case, for permutation  $(p_1, p_2) = (2, 1)$ , Jesse can mark  $p_1$  yellow and  $p_2$  blue. After sorting it will remain being  $(2, 1)$ , with score 0.

In the second test case, for permutation  $(p_1, p_2, p_3, p_4) = (2, 1, 4, 3)$ , Jesse can mark  $p_1, p_2$  yellow and  $p_3, p_4$  blue. After sorting the permutation will become  $(1, 2, 3, 4)$ , with score 4.

## Problem K. Knocker

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

*I am the danger. A guy opens  
his door and gets shot, and you  
think that of me? No. I am the  
one who knocks!*

—Walter White, *Breaking Bad*

It's well-known that Walter is the one who knocks. But did you know that he can knock with strength  $x$  for every positive integer  $x$ ?

He has an array  $a_1, a_2, \dots, a_n$  of positive integers. If he knocks with strength  $x$ , then, for each  $1 \leq i \leq n$ ,  $a_i$  will be replaced with  $a_i \bmod x$ .

How many different arrays  $a_1, a_2, \dots, a_n$  can Walter achieve by knocking any number of times? As the number can be very large, output it modulo 998244353.

Here  $x \bmod y$  denotes the remainder of  $x$  when dividing by  $y$ . For example,  $6 \bmod 3 = 0$ , and  $6 \bmod 4 = 2$ .

### Input

The first line of the input contains a single integer  $n$  ( $1 \leq n \leq 500$ ) — the length of the array.

The second line of the input contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 500$ ) — the elements of the array.

### Output

Output a single integer — the number of different arrays Walt can achieve by knocking.

### Examples

standard input	standard output
1 5	4
2 6 5	7
5 1 2 4 8 16	69

### Note

In the first sample, you can get the following arrays:  $[5], [2], [1], [0]$ .

In the second sample, you can get the following arrays:  $[6, 5], [2, 1], [1, 0], [0, 5], [0, 2], [0, 1], [0, 0]$ .



## Problem L. Lalo's Lawyer Lost

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

*Jesus, it's always the desert.*

—James McGill, *Better Call Saul*

While picking up Lalo's bail, Saul and Mike got lost in a desert. In a desert, they found a cactus: a connected undirected graph on  $n$  nodes, such that every edge in it appears in at most one simple cycle.

It turned out that  $n$  is even. For some reason, Mike and Saul want to solve the following problem:

Define  $d(u, v)$  as the shortest distance between nodes  $u$  and  $v$  in this cactus. Partition all  $n$  nodes into  $\frac{n}{2}$  pairs  $(u_i, v_i)$ , such that every node appears in exactly one pair, and the sum of  $d(u_i, v_i)$  is **maximized**.

What's the maximum possible sum of  $d(u_i, v_i)$  in such a partition?

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 10^5$ ) — the number of test cases. The description of test cases follows.

The first line of each test case contains two integers  $n, m$  ( $2 \leq n \leq 2 \cdot 10^5, n - 1 \leq m \leq 4 \cdot 10^5, n$  is even) — the number of nodes and edges correspondingly.

The  $i$ -th of the next  $m$  lines contains two integers  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i$ ) — denoting an edge between nodes  $u_i$  and  $v_i$ . It's guaranteed that these edges form a cactus.

It is guaranteed that the sum of  $n$  over all test cases does not exceed  $2 \cdot 10^5$ , and the sum of  $m$  over all test cases does not exceed  $4 \cdot 10^5$ .

### Output

For each test case, output a single integer — the largest possible sum of  $d(u_i, v_i)$  in such a partition.

## Example

standard input	standard output
3	4
4 3	7
1 2	11
2 3	
3 4	
6 7	
1 2	
2 3	
3 1	
3 4	
4 5	
5 6	
6 4	
8 9	
1 2	
2 3	
3 1	
3 4	
4 5	
5 6	
6 7	
7 8	
8 3	