Problem A. Welcome

Time limit: 1 second
Memory limit: 512 megabytes

Welcome to ACM! Let's start with a simple game. Just pick a number. If it's even, say "Hello ACM," and if it's odd, say "Goodbye ACM."

Input

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

then followed by t testcases. Each consists of a single integer like z were z is $1 \leq z \leq 1000$.

Output

For each day, print one integer - the number of scores Morqi will have at the end of that day.

Examples

test	answer
3 1 4 3	Goodbye ACM Hello ACM Goodbye ACM

Problem B. Regular Polygon

Time limit:	0.5 seconds
Memory limit:	256 megabytes

Ali needs a fence around his farm but is too lazy to build it himself. So he purchased a fence-building robot.

He wants the fence to be a regular polygon. The robot builds the fence along a single path, but it can only make fence corners at a single angle a.

Will the robot be able to build the fence Ali wants? In other words, is there a regular polygon whose angles are equal to a?

Input

The first line of input contains an integer t $(0 < t < 180)$: the number of tests. Each of the following t lines contains a single integer a $(0 < a < 180)$: the angle the robot can make corners at measured in degrees.

Output

For each test, output on a single line "YES" (without quotes), if the robot can build a fence Ali wants, and "NO" (without quotes), if it is impossible.

Examples

test	answer
3 30 60 90	NO YES YES

Explanations

In the first test case, it is impossible to build the fence, since there is no regular polygon with angle 30°.

In the second test case, the fence is a regular triangle, and in the last test case: a square.

Problem C. Wandering in Japan

Time limit: 2 seconds
Memory limit: 1024 megabytes

After enduring days of relentless penalty kicks with Morqi, Gavi had an epiphany: soccer was more than just penalties. With newfound clarity, he resolved to return to Earth and journey to Japan to seek out the legendary captain Tsubasa for training. However, upon his arrival in Japan, hunger gnawed at his stomach, distracting him from thoughts of soccer. Empty of yens to purchase food, he sought his old friend, Pishi.

Pishi brought him some delicious Dangos but has put them in a non-common order. Since Gavi is a professional dango eater, he only eats his Dangos when served traditionally. A traditional Dango of length L is of length $L+1$ and contains L Dangos and a piece of stick. For example, a traditional Dango of length 3 might look like "-ooo" or "ooo-". Since Gavi is so hungry, he wonders how can he break the main stick so that he can obtain the longest traditional Dango sequence.

Input

The first line contains a single integer $n (1 \leq n \leq 2 \times 10^5)$ - length of the dango Pishi brought.

The second line contains a string of 'o' and '-' with length n - represents the dango Pishi brought. we represent the dango with "o" and the stick with "-"

Output

A single integer - the largest sequence of traditional dango Gavi can eat. If no sequence creates a traditional dango, print -1.

Examples

test	answer
10 o-oooo---o	4
1 -	-1
30 -o-o-oooo-oo-o-ooooooo--oooo-o	7

Problem D. Stammer

Time limit: 0.5 seconds
Memory limit: 256 megabytes

One of Erfans friends has a stutter(a speech disorder). He usually repeats or adds some syllables to a word. So Erfan sometimes doesn't know if he got his word correctly. So he wants you to write a program that tells him if we can build the word he heard by deleting some of the syllables that his friend said. for example, we can build the word "ACM" from the word "AaaCbnM" notice that 'A' differs from 'a'. you should print YES if the word can be built, and otherwise NO.

Input

The first line contains one integer, t $(1 \leq t \leq 10^5)$ - the number of test cases. each t next line contains two strings. The first is the word Erfan thinks his friend means and the second is the word his friend said. The length of the strings will not exceed 100.

Output

Print a single string for each test case containing YES or NO.

Examples

test	answer
2 hello hhheeelo hello heeflldooo	NO YES
1 salam salam	YES
4 ACM acccmmmAacmCM ACMM ACMededmmm hi hey wordddd wefewfeefwefewfeworedefewfdfdfdfdf	YES NO NO YES

Problem E. Reisi Revenge

Time limit: 1 second
Memory limit: 256 megabytes

Ali and Mohammad are good friends, but in the second semester of the year Ali pranked Mohammad by deleting two subjects in the "Tarmim". Mohammad doesn't start a fight, but instead, he wants to revenge.

Mohammad needs Ali's GOLESTAN password so he can delete his whole semester. Mohammad found some passwords in his chat with Ali. He wants to see if any of them are Ali's GOLESTAN password. However, he can only try one password, or he'll get blocked. So you will help Mohammad to see if he got lucky and got Ali's password.

Mohammad knows two things about Ali's password:

1- Ali's password is pangram which means that it contains all the English letters at least once.
2- Ali likes the number one, so when we arrange the numbers in his password (remaining in the same sequence as it was in the string) , we can divide them into two equal parts (in length) and we only need exactly ONE operation to equalize the value of the two parts (the operations will be explained later).

The operations can be executed as described above. Suppose you have two positive integers a and b.

In one move, you can change a in the following way:

- Choose any positive odd integer x ($x > 0$) and replace a with $a + x$;
- Choose any positive even integer y ($y > 0$) and replace a with $a - y$.

You need to check if Mohammad got lucky enough and found Ali's password, by answering t independent test cases.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

then followed by t testcases. Each consists of a string with a maximum length of 100 characters. **it's guaranteed that every password has even numeric digits, and the numbers can fit in an integers variables :)**

Output

For each test case, output on a single line "Iucky" (without quotes), if the password might be Ali's password, and "not lucKy" (without quotes) if it is not.

Examples

test	answer
3 IHATERIESI13 1asdfghjklqwertyuiopzxcvbnm4 TheQuickBrownFox4JumpsOverTheLazyDog2	not lucKy Iucky Iucky

Explanations

In the third test case, the string has all English letters and when we arrange the numbers with each other we get 42 so now we have $a = 4$ and $b = 2$ and when we subtract a with the even number 2 we can get b.

Problem F. Magic Strawberries

Time limit: 1 second
Memory limit: 256 megabytes

Two little hobbits, Pippin and Merry, are lost in the jungle, they have x strawberries to eat, but the strawberries are magical and hobbits can change the number of them by two following rules:

1. if x is even, they can turn it into $\frac{3a}{2}$.
2. if x is greater than one, they can turn it into $a - 1$.

They are too captious and they want to have y strawberries. they want to see if they can obtain y from x using the rules they know. The rules can be used any number of times in any order. It is not required to use rules, they can leave x as it is.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$), the number of test cases.

The only line of each test case contains two integers x and y ($1 \leq x, y \leq 10^9$) — the current number of strawberries and the number that they want to get.

Output

For the i-th test case print YES if they can get the number y from the number x using known rules, and NO otherwise

Examples

test	answer
7 2 3 1 1 3 6 6 8 1 2 4 1 31235 6578234	YES YES NO YES NO YES YES

Problem G. Cookie Game

Time limit: 1 second
Memory limit: 256 megabyte

Mahsa and Negar are playing the Cookie Game. The game has 10^{100} turns. There are n boxes, and the i-th box has a_i cookies. in each turn Mahsa chooses one of the boxes and eats x cookies from it, if there are less than x cookies in the box, she eats all of it. Then Negar chooses a box and adds y cookies to it, Negar cannot choose an empty box.

Mahsa's goal is to maximize the number of empty boxes and Negar's goal is to minimize it. What is the final number of empty boxes if both play optimally?

Input

The first line contains three positive integers n, x, and y ($1 \le n \le 100$, $1 \le x, y \le 10^5$) — the number of boxes, value x and value y, respectively.

The second contains n integers a_1, a_2, ..., a_n ($1 \le a_i \le 100$), where a_i is the initial number of cookies in the i-th box.

Output

Print the number of empty boxes at the end of the game, if both of them play optimally.

Examples

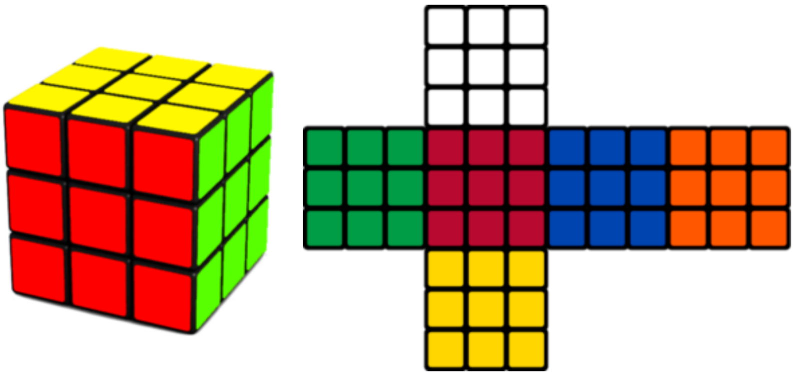
test	answer
6 3 2 2 3 1 3 4 2	6
5 5 6 1 2 6 10 3	2

Problem H. Rubik Coloring

Time limit:	2 seconds
Memory limit:	256 megabytes

You have a perfect binary tree of $2^k - 1$ nodes - a binary tree where all vertices i from 1 to $2^{k-1} - 1$ have exactly two children: vertices $2i$ and $2i + 1$. Vertices from 2^{k-1} to $2^k - 1$ don't have any children. You want to color its vertices with the 6 Rubik's cube colors (White, Green, Red, Blue, Orange, and Yellow).

Let's call a coloring *good* when all edges connect nodes with colors that are *neighboring* sides in the Rubik's cube.



More formally:

- A white node can not be neighboring with white and yellow nodes.
- A yellow node can not be neighboring with white and yellow nodes.
- A green node can not be neighboring with green and blue nodes.
- A blue node can not be neighboring with green and blue nodes.
- A red node can not be neighboring with red and orange nodes.
- An orange node can not be neighboring with red and orange nodes.

You want to calculate the number of good colorings of the binary tree. Two colorings are considered different if at least one node is colored differently.

The answer may be too large, so output the answer modulo $10^9 + 7$.

Input

The first and only line contains the integers $k (1 \le k \le 60)$ - the number of levels in the perfect binary tree you need to color.

Output

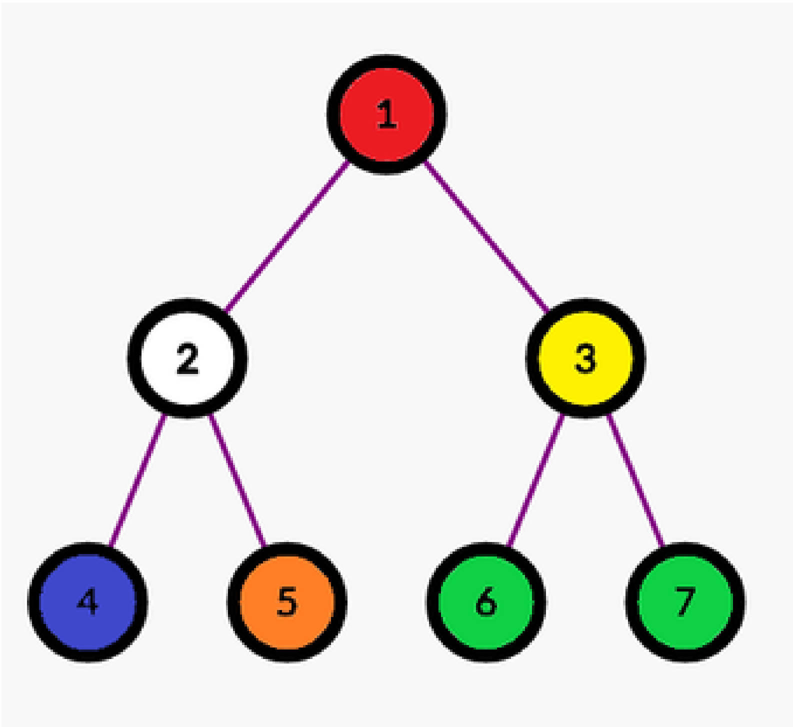
Print one integer - the number of the different colorings modulo $10^9 + 7$.

Examples

test	answer
3	24576
14	934234

Explanations

In the picture below, you can see one of the correct colorings of the first example.



Problem I. Obsession

Time limit: 1 second
Memory limit: 256 megabytes

Erfan has an obsession with having the most equal numbers in an array. He always wants to find a way to make numbers equal as many numbers as he can. but just making them equal doesn't seem to be much fun. So he decided to make a game. suppose he has an array a of size n. he can create another array b containing a permutation of 1 to n and then build the final array by adding corresponding numbers in these arrays to each other. his goal is to make the most equal numbers he can. for example for $a = [1, 2, 2]$ one of the optimal answers is $b = [2, 1, 3]$ so the final array will be $[3, 3, 5]$. and has 2 equal numbers. you must find the maximum number of elements he can make equal.

Permutation of 1 to n is a sequence of numbers which contains every number from 1 to n exactly once.

Input

The first line contains t $(1 \le t \le 10^4)$ - the number of test cases. the first line of each test case contains one integer n $(1 \le n \le 10^5)$ - the length of Erfans array. the second line of each test case contains n integers a_1 to a_n $(0 \le a_i \le 10^9)$.

Output

For each test case print the maximum number of elements that can be made equal.

Examples

test	answer
5 3 1 2 2 5 1 2 3 4 5 4 1 2 101 102 3 102 1033 10 5 1 101 102 103 1	2 5 2 1 3

Explanations

For the last test case, the permutation we use can be $[1, 5, 4, 3, 2]$.

Problem J. The Depressed Hacker

Time limit: 1 second
Memory limit: 256 megabytes

"Welcome to the Prison" This is the first thing I saw when I hacked into the prison database!

but there is a small problem...

let me tell you a story first. Once, he was a good guy living peacefully with his pretty wife, but not until his wife got cancer and needed special treatment, the problem was when the guy knew the treatment's price. At first, he realized that he needed to do something to get this amount of money, but his wife had only 2 weeks to start the treatment or she would die.

As we all know the fastest way to earn money is by committing a crime. Our guy was a computer engineer, so he started his career as a computer hacker. He used his skills for the Italian mafia, and after a big bank robbery that he made with them, he waited
and waited and waited...
They ignored him and he got no money.
and his wife...

Two years passed, and the guy was living depressed and lonely, but the thing that kept him alive was that he heard that the mafia boss was in prison right now. So he hired a serial killer to kill him in prison. The serial killer told the guy that he only needed the ID of the mafia boss. So the guy hacked into the prison database and realized two things:

1- The IDs are 8 blocks separated with ":" where each block is 4 hexadecimal characters (a56f:00d3:0000:0124:0001:0000:0000:0000)

2- The more important thing is that they had to shorten the IDs in the database to take less space. But I need the full ID of the mafia boss to give it to the serial killer. AND YOU WILL HELP ME WITH IT!

The record of an ID address can be shortened by removing one or more leading zeroes at the beginning of each block. However, each block should contain at least one digit in the short format. For example, the leading zeroes can be removed like that: "a56f:00d3:0000:0124:0001:f19a:1000:0000" → "a56f:d3:0:0124:01:f19a:1000:00". There are more ways to shorten zeroes in this ID.
Some IDs contain long sequences of zeroes. Continuous sequences of zero blocks can be shortened to "::". A sequence can consist of one or several consecutive blocks, with all characters equal to 0. You can see examples of zero-block shortenings below:

"a56f:00d3:0000:0124:0001:0000:0000:0000" → "a56f:00d3:0000:0124:0001::";
"a56f:0000:0000:0124:0001:0000:1234:0ff0" → "a56f::0124:0001:0000:1234:0ff0";
"a56f:0000:0000:0000:0001:0000:1234:0ff0" → "a56f:0000::0000:0001:0000:1234:0ff0";
"a56f:00d3:0000:0124:0001:0000:0000:0000" → "a56f:00d3:0000:0124:0001::0000";
"0000:0000:0000:0000:0000:0000:0000:0000" → "::"

It is not allowed to shorten zero blocks in the address more than once. This means that the short record can't contain the sequence of characters "::" more than once. Otherwise, it will sometimes be impossible to determine the number of zero blocks, each represented by a double colon.

You've got several shortened records of IDs. Restore their full record.

Input

The first line contains a single positive integer t $(1 \leq t \leq 100)$ — the number of test cases.

Each of the following n lines contains a string — the short ID. Each string only consists of string characters "0123456789abcdef:".

It is guaranteed that each short address is obtained in the way that is described in the statement from some full ID.

Output

For each short ID from the input print its full record on a separate line. Print the full records for the short ID in the order, in which the short records follow in the input.

Examples

test	answer
6 a56f:d3:0:0124:01:f19a:1000:00 a56f:00d3:0000:0124:0001:: a56f::0124:0001:0000:1234:0ff0 a56f:0000::0000:0001:0000:1234:0ff0 :: 0ea::4d:f4:6:0	a56f:00d3:0000:0124:0001:f19a:1000:0000 a56f:00d3:0000:0124:0001:0000:0000:0000 a56f:0000:0000:0124:0001:0000:1234:0ff0 a56f:0000:0000:0000:0001:0000:1234:0ff0 0000:0000:0000:0000:0000:0000:0000:0000 00ea:0000:0000:0000:004d:00f4:0006:0000

Problem K. Powerful Numbers

Time limit: 3 seconds
Memory limit: 256 megabytes

A number is called *powerful* if it is a power of two or a factorial. In other words, the number m is powerful if there exists a non-negative integer d such that $m = 2^d$ or $m = d!$, where $d! = 1 \times 2 \times \cdots \times d$ (in particular, $0! = 1$). For example 1, 4, and 6 are powerful numbers, because $1 = 1!$, $4 = 2^2$, and $6 = 3!$ but 7, 10, or 18 are not.

You are given a positive integer n. Find the minimum number k such that n can be represented as the sum of k **distinct** powerful numbers, or say that there is no such k.

Input

The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

A test case consists of only one line, containing one integer n ($1 \le n \le 10^{12}$).

Output

For each test case print the answer on a separate line.

If n can not be represented as the sum of **distinct** powerful numbers, print 1.

Otherwise, print a single positive integer — the minimum possible value of k.

Examples

test	answer
4 7 11 240 17179869184	2 3 4 1

Explanations

In the first test case, 7 can be represented as $7 = 1 + 6$, where 1 and 6 are powerful numbers. Because 7 is not a powerful number, we know that the minimum possible value of k in this case is $k = 2$.

In the second test case, a possible way to represent 11 as the sum of three powerful numbers is $11 = 1+4+6$. We can show that there is no way to represent 11 as the sum of two or less powerful numbers.

In the third test case, 240 can be represented as $240 = 24 + 32 + 64 + 120$. Observe that $240 = 120 + 120$ is not a valid representation, because the powerful numbers have to be **distinct**.

In the fourth test case, $17179869184 = 2^{34}$, so 17179869184 is a powerful number and the minimum k in this case is $k = 1$.

Problem L. Leopardo Gavinchi

Time limit: 2 seconds
Memory limit: 1024 megabytes

After eating lots of dango, Gavi became fat and couldn't play soccer like before anymore. So he returned to his hometown Italy and considered taking up his ancestors' profession, painting. Everybody knows Italy has a reputation for art-zealous leopards!

Italian leopards like certain kinds of art. If you decompose the painting into a $N \times M$ grid and give each cell of this grid either a high-saturated or a low-saturated color, the art would meet the Italian Leopards standard and, thus be considered beautiful, if the following condition is satisfied:

- If square (i, j) is high saturated then squares $(i + 1, j)$ and $(i + 1, j + 1)$ should also be high saturated(if they exist).

An Italian leopard has given Gavi a painting and asked to make it beautiful only by increasing the saturation of some(possibly zero) squares(from low to high). Now Gavi asks you, how many ways he can do that.

Input

First line comes N and M, lengths of the painting grid $1 \leq N, M \leq 2000$

After that, there are N lines, each line having a string S_i of length M consisting of `.` and `#` - showing the saturation of the painting in that square, `.` being low saturated and `#` for high saturation.

Output

Print the number of different girds Gavi can create, only by changing some low-saturated colors to high-saturated ones. Two grids are considered different if at least one square in which the saturations differ.

Print out the answer modulo 998244353, since Gavi can't read big numbers.

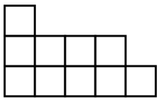
Examples

test	answer
2 2 .# ..	3
5 5 ....# ...#. ..#.. .#.#. #...#	92
25 25	604936632

Problem M. Maximum Domino

Time limit: 0.5 seconds
Memory limit: 256 megabytes

You are given a histogram with n columns of length $a_1, a_2, \ldots, a_n$. ($a_1 \geq a_2 \geq \ldots \geq a_n \geq 1$)



Histogram for $a = [3, 2, 2, 2, 1]$.

Your goal is to find the largest number of non-overlapping dominos that you can draw inside of this histogram, a domino is a 1×2 or 2×1 rectangle

Input

The first line of input contains a single integer t ($1 \leq t \leq 10\,000$): the number of test cases. Then the descriptions of the test cases follow.

The first line of each test case contains one integer $n (1 \leq n \leq 100\,000)$: the number of columns in the given histogram.

The next line of each test case contains n integers $a_1, a_2, , a_n$ $(1 \leq a_i \leq 300000, a_i \geq a_{i+1})$: the lengths of columns.

It is guaranteed that the sum of the values of n for all test cases in a test does not exceed $100\,000$.

Output

For each test case output one integer: the largest number of non-overlapping dominos you can draw inside the given histogram.

Examples

test	answer
2 5 3 2 2 2 1 3 3 3 3	4 4

Explanations

Some of the possible solutions for the first test case of the example: