

Problem A. Vanya and Primes

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

Schoolboy Vanya recently learned prime numbers. His dream is to find the "last" prime in the universe. Let's help him move towards his dream and find n-th prime for him.

Input

Integer n ($1 \leq n \leq 1000$)

Output

n-th prime number

Examples

standard input	standard output
1	2
2	3
5	11
8	19

Note

You must use function.

Problem B. There is an impostor among us

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

There is an impostor among us. In the first minutes of every contest he submits his solutions at the speed of a machine-gun. Besides, he cheats by means of using several ip addresses for submitting. You already know that cheating is prohibited in KBTU. So, we decided to disqualify the cheaters. We know who submitted their solutions in the first minutes of a contest. For this we need to analyze the list of submits. If a person has submitted 2 or more solutions, he is a cheater.

Input

The first line contains the number N ($3 \leq N \leq 100$) of submissions in the first minutes of the contest. The next N lines contain the string - name of student who submitted solution.

Output

The name of the cheater in each line. The order should be as they occur in the input list. If there are no cheaters, output: "Understandable, have a great day" without quotes.

Examples

standard input	standard output
5 Sanzhar Yato Yato Giyu Sanzhar	Yato Sanzhar
7 Sanzhar Yato Giyu Giyu Yato Giyu Sanzhar	Giyu Yato Sanzhar
4 Kanao Tanjiro Mitsuri Obanai	Understandable, have a great day
6 Sasuke Naruto Sasuke Naruto Sasuke Naruto	Sasuke Naruto

Note

Don't sort names.

Problem C. S - Scanner

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

The Yorha unit 9S wants to scan the area for Yorha unit 2B. They landed at x, y coordinates on the area with $N \times N$ units. Row and column numeration have started from 1 at the left upper corner.

There are two types of zones in this area. An inaccessible cell where they cannot move is marked as '*'. The cell where they can move is marked as '.'.

9S must determine how many free cells (4 neighbor cells) there are around them in order to successfully complete the task. 9S and 2B always land on accessible space. The boundaries of the territory are always inaccessible for movement.

Input

First line: Integer N ($3 \leq N \leq 10$), length and width of the territory.

Second line: Integers x and y . ($2 \leq x, y \leq N - 1$), where x - column index, y - row index.

Next N lines: N chars in a row, consisting of '.' or '*' without quotes.

Output

Integer - how many cells are available.

Examples

standard input	standard output
4 2 3 **** **.* *.* ****	1
6 4 2 ***** ***.* ***.* **.* **.* *****	3
8 3 3 ***** **.*.* *.*.* **.*.* *.*.* *.*.* *.*.* *****	7

Note

You must use recursion.

Problem D. Bonapity

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

A group of junior programmers are attending an advanced programming camp, where they learn very difficult algorithms and programming techniques! Near the center in which the camp is held, is a professional bakery which makes tasty pastries and pizza. It is called 'Bonabity'... or 'Ponapety'... or 'Ponabity'... Actually no one knows how to spell this name in English, even the bakery owner doesn't, and the legends say that Arabs always confuse between 'b' and 'p', and also between 'i' and 'e', so 'b' for them is just the same as 'p', and 'i' for them is just the same as 'e', they also don't care about letters' cases (uppercase and lowercase for a certain letter are similar). For example, the words 'Ponabity' and 'bonabety' are considered the same. You are given two words including only upper case and lower case English letters, and you have to determine whether the two words are similar in Arabic.

Input

The input consists of several test cases. The first line of the input contains a single integer T, the number of the test cases. Each of the following T lines represents a test case and contains two space-separated strings (each one consists of only upper case and lower case English letters and its length will not exceed 100 characters).

Output

For each test case print a single line: 'Yes' if the words are similar in Arabic and 'No' otherwise.

Examples

standard input	standard output
4	Yes
Ponabity bonabety	Yes
barbie barpee	No
abcabc apcap	No
abc apcd	
5	Yes
Aba aba	Yes
TEST test	Yes
TEST TesT	Yes
FUNction funCTION	No
AA BB	

Note

Solve problem creating your own **functions**.

Problem E. Unique array

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

You are given an array with size - N .

Your task is - to determine the uniqueness of an array.

Input

In the first line given N - the size of the array.

In the next line given elements.

Output

Print *YES* if the given array contains only unique elements, otherwise print *NO*.

Examples

standard input	standard output
7 2 4 3 -1 7 12 -4	YES
5 5 2 -3 2 1	NO

Note

Use STL container - set.

Problem F. Prime Factorization

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

You're given integer N , write a **function** that will return prime factorization of this integer.

Input

Single line, containing integer N , ($2 \leq N \leq 10^8$).

Output

Output prime factorization of N in ascending order, each prime divided by space.

Examples

standard input	standard output
8	2 2 2
47	47
78	2 3 13

Note

Prime factorization - is finding which prime numbers multiply together to make the original number.

Example - prime factors of 12?

$$2 * 2 * 3 = 12$$

Problem G. Prime Factorization v.2

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

You're given integer N , write **recursion** that will return prime factorization of N .

Input

Single line, containing integer N , ($2 \leq N \leq 10^8$).

Output

Output prime factorization of N , each prime divided by space in ascending order.

Examples

standard input	standard output
8	2 2 2
47	47
78	2 3 13
14	2 7

Note

"Prime Factorization" is finding which prime numbers multiply together to make the original number.