Problem A. Two Strings

OS Linux

Given two strings, determine if they share a common substring. A substring may be as small as one character.

Example
$s1 =$ 'and'
$s2 =$ 'art'

These share the common substring a.

$s1 =$ 'be'
$s2 =$ 'cat'

These do not share a substring.

Function Description

Complete the function *twoStrings* in the editor below.

twoStrings has the following parameter(s):

- *string s1:* a string
- *string s2:* another string

Returns

- *string:* either `YES` or `NO`

Input Format

The first line contains a single integer p, the number of test cases.

The following p pairs of lines are as follows:

- The first line contains string $s1$.
- The second line contains string $s2$.

Constraints

- $s1$ and $s2$ consist of characters in the range ascii[a-z].
- $1 \leq p \leq 10$
- $1 \leq |s1|, |s2| \leq 10^5$

Output Format

For each pair of strings, return `YES` or `NO`.

Input	Output
2 hello world hi world	YES NO

Explanation

We have $p = 2$ pairs to check:

1. $s1 =$ **"hello"**, $s2 =$ **"world"**. The substrings **"o"** and **"l"** are common to both strings.
2. $a =$ **"hi"**, $b =$ **"world"**. $s1$ and $s2$ share no common substrings.

Problem B. Boy or Girl

Time Limit	1000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

Those days, many boys use beautiful girls' photos as avatars in forums. So it is pretty hard to tell the gender of a user at the first glance. Last year, our hero went to a forum and had a nice chat with a beauty (he thought so). After that they talked very often and eventually they became a couple in the network.

But yesterday, he came to see "her" in the real world and found out "she" is actually a very strong man! Our hero is very sad and he is too tired to love again now. So he came up with a way to recognize users' genders by their user names.

This is his method: if the number of distinct characters in one's user name is odd, then he is a male, otherwise she is a female. You are given the string that denotes the user name, please help our hero to determine the gender of this user by his method.

Input

The first line contains a non-empty string, that contains only lowercase English letters — the user name. This string contains at most 100 letters.

Output

If it is a female by our hero's method, print `"CHAT WITH HER!"` (without the quotes), otherwise, print `"IGNORE HIM!"` (without the quotes).

Examples

Input	Output
`wjmzbmr`	`CHAT WITH HER!`

Input	Output
xiaodao	IGNORE HIM!

Input	Output
sevenkplus	CHAT WITH HER!

Note

For the first example. There are 6 distinct characters in "wjmzbmr". These characters are: "w", "j", "m", "z", "b", "r". So wjmzbmr is a female and you should print "CHAT WITH HER!".

Problem C. Registration System

Time Limit	5000 ms
Mem Limit	65536 kB
Input File	stdin
Output File	stdout

A new e-mail service "Berlandesk" is going to be opened in Berland in the near future. The site administration wants to launch their project as soon as possible, that's why they ask you to help. You're suggested to implement the prototype of site registration system. The system should work on the following principle.

Each time a new user wants to register, he sends to the system a request with his `name`. If such a `name` does not exist in the system database, it is inserted into the database, and the user gets the response `OK`, confirming the successful registration. If the `name` already exists in the system database, the system makes up a new user name, sends it to the user as a prompt and *also inserts the prompt into the database*. The new name is formed by the following rule. Numbers, starting with 1, are appended one after another to `name` (`name1`, `name2`, ...), among these numbers the least i is found so that `name`i does not yet exist in the database.

Input

The first line contains number n ($1 \le n \le 10^5$). The following n lines contain the requests to the system. Each request is a non-empty line, and consists of not more than 32 characters, which are all lowercase Latin letters.

Output

Print n lines, which are system responses to the requests: `OK` in case of successful registration, or a prompt with a new name, if the requested name is already taken.

Examples

Input	Output
4 abacaba acaba abacaba acab	OK OK abacaba1 OK

Input	Output
6 first first second second third third	OK first1 OK second1 OK third1

Problem D. Anton and Letters

Time Limit	2000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

Recently, Anton has found a set. The set consists of small English letters. Anton carefully wrote out all the letters from the set in one line, separated by a comma. He also added an opening curved bracket at the beginning of the line and a closing curved bracket at the end of the line.

Unfortunately, from time to time Anton would forget writing some letter and write it again. He asks you to count the total number of distinct letters in his set.

Input

The first and the single line contains the set of letters. The length of the line doesn't exceed 1000. It is guaranteed that the line starts from an opening curved bracket and ends with a closing curved bracket. Between them, small English letters are listed, separated by a comma. Each comma is followed by a space.

Output

Print a single number — the number of distinct letters in Anton's set.

Examples

Input	Output
{a, b, c}	3

Input	Output
{b, a, b, a}	2

Input	Output
{}	0

Problem E. Chat Order

Time Limit 3000 ms
Mem Limit 262144 kB

Polycarp is a big lover of killing time in social networks. A page with a chatlist in his favourite network is made so that when a message is sent to some friend, his friend's chat rises to the very top of the page. The relative order of the other chats doesn't change. If there was no chat with this friend before, then a new chat is simply inserted to the top of the list.

Assuming that the chat list is initially empty, given the sequence of Polycaprus' messages make a list of chats after all of his messages are processed. Assume that no friend wrote any message to Polycarpus.

Input

The first line contains integer n ($1 \le n \le 200\,000$) — the number of Polycarpus' messages. Next n lines enlist the message recipients in the order in which the messages were sent. The name of each participant is a non-empty sequence of lowercase English letters of length at most 10.

Output

Print all the recipients to who Polycarp talked to in the order of chats with them, from top to bottom.

Examples

Input	Output
4 alex ivan roman ivan	ivan roman alex

Input	Output
8 alina maria ekaterina darya darya ekaterina maria alina	alina maria ekaterina darya

Note

In the first test case Polycarpus first writes to friend by name "alex", and the list looks as follows:

1. alex

Then Polycarpus writes to friend by name "ivan" and the list looks as follows:

1. ivan
2. alex

Polycarpus writes the third message to friend by name "roman" and the list looks as follows:

1. roman
2. ivan
3. alex

Polycarpus writes the fourth message to friend by name "ivan", to who he has already sent a message, so the list of chats changes as follows:

1. ivan
2. roman
3. alex

Problem F. Keyboard

Time Limit	2000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

Our good friend Mole is trying to code a big message. He is typing on an unusual keyboard with characters arranged in following way:

```
qwertyuiop
asdfghjkl;
zxcvbnm,./
```

Unfortunately Mole is blind, so sometimes it is problem for him to put his hands accurately. He accidentally moved both his hands with one position to the left or to the right. That means that now he presses not a button he wants, but one neighboring button (left or right, as specified in input).

We have a sequence of characters he has typed and we want to find the original message.

Input

First line of the input contains one letter describing direction of shifting (`'L'` or `'R'` respectively for left or right).

Second line contains a sequence of characters written by Mole. The size of this sequence will be no more than 100. Sequence contains only symbols that appear on Mole's keyboard. It doesn't contain spaces as there is no space on Mole's keyboard.

It is guaranteed that even though Mole hands are moved, he is still pressing buttons on keyboard and not hitting outside it.

Output

Print a line that contains the original message.

Examples

Input	Output
R s;;upimrrfod;pbr	allyouneedislove

Problem G. Potions (Hard Version)

Time Limit 1000 ms
Mem Limit 262144 kB

This is the hard version of the problem. The only difference is that in this version $n \leq 200000$. You can make hacks only if both versions of the problem are solved.

There are n potions in a line, with potion 1 on the far left and potion n on the far right. Each potion will increase your health by a_i when drunk. a_i can be negative, meaning that potion will decrease will health.

You start with 0 health and you will walk from left to right, from first potion to the last one. At each potion, you may choose to drink it or ignore it. **You must ensure that your health is always non-negative**.

What is the largest number of potions you can drink?

Input

The first line contains a single integer n ($1 \leq n \leq 200000$) — the number of potions.

The next line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \leq a_i \leq 10^9$) which represent the change in health after drinking that potion.

Output

Output a single integer, the maximum number of potions you can drink without your health becoming negative.

Examples

Input	Output
6 4 -4 1 -3 1 -3	5

Note

For the sample, you can drink 5 potions by taking potions 1, 3, 4, 5 and 6. It is not possible to drink all 6 potions because your health will go negative at some point

Problem H. Heap Operations

Time Limit 1000 ms
Mem Limit 262144 kB

Petya has recently learned data structure named "Binary heap".

The heap he is now operating with allows the following operations:

- put the given number into the heap;
- get the value of the minimum element in the heap;
- extract the minimum element from the heap;

Thus, at any moment of time the heap contains several integers (possibly none), some of them might be equal.

In order to better learn this data structure Petya took an empty heap and applied some operations above to it. Also, he carefully wrote down all the operations and their results to his event log, following the format:

- `insert` x — put the element with value x in the heap;
- `getMin` x — the value of the minimum element contained in the heap was equal to x;
- `removeMin` — the minimum element was extracted from the heap (only one instance, if there were many).

All the operations were correct, i.e. there was at least one element in the heap each time `getMin` or `removeMin` operations were applied.

While Petya was away for a lunch, his little brother Vova came to the room, took away some of the pages from Petya's log and used them to make paper boats.

Now Vova is worried, if he made Petya's sequence of operations inconsistent. For example, if one apply operations one-by-one in the order they are written in the event log, results of `getMin` operations might differ from the results recorded by Petya, and some of `getMin` or `removeMin` operations may be incorrect, as the heap is empty at the moment they are applied.

Now Vova wants to add some new operation records to the event log in order to make the resulting sequence of operations correct. That is, the result of each `getMin` operation is equal to the result in the record, and the heap is non-empty when `getMin` ad `removeMin` are applied. Vova wants to complete this as fast as possible, as the Petya may get back at

any moment. He asks you to add the least possible number of operation records to the current log. Note that arbitrary number of operations may be added at the beginning, between any two other operations, or at the end of the log.

Input

The first line of the input contains the only integer n ($1 \le n \le 100\,000$) — the number of the records left in Petya's journal.

Each of the following n lines describe the records in the current log in the order they are applied. Format described in the statement is used. All numbers in the input are integers not exceeding 10^9 by their absolute value.

Output

The first line of the output should contain a single integer m — the minimum possible number of records in the modified sequence of operations.

Next m lines should contain the corrected sequence of records following the format of the input (described in the statement), one per line and in the order they are applied. All the numbers in the output should be integers not exceeding 10^9 by their absolute value.

Note that the input sequence of operations must be the **subsequence** of the output sequence.

It's guaranteed that there exists the correct answer consisting of no more than $1\,000\,000$ operations.

Examples

Input	Output
2 insert 3 getMin 4	4 insert 3 removeMin insert 4 getMin 4

Input	Output
4 insert 1 insert 1 removeMin getMin 2	6 insert 1 insert 1 removeMin removeMin insert 2 getMin 2

Note

In the first sample, after number 3 is inserted into the heap, the minimum number is 3. To make the result of the first `getMin` equal to 4 one should firstly remove number 3 from the heap and then add number 4 into the heap.

In the second sample case number 1 is inserted two times, so should be similarly removed twice.

Problem I. Sorting Queries

Time Limit 2000 ms
Mem Limit 1048576 kB

Problem Statement

We have an empty sequence A. You will be given Q queries, which should be processed in the order they are given. Each query is of one of the three kinds below:

- `1 x` : Append x to the end of A.
- `2` : Print the element at the beginning of A. Then, delete that element. It is guaranteed that A will not empty when this query is given.
- `3` : Sort A in ascending order.

Constraints

- $1 \leq Q \leq 2 \times 10^5$
- $0 \leq x \leq 10^9$
- A will not be empty when a query `2` is given.
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

$$
\begin{aligned}
&Q \\
&\mathrm{query}1 \\
&\mathrm{query}2 \\
&\vdots \\
&\mathrm{query}Q
\end{aligned}
$$

The i-th query, query$i$, begins with the kind of query c_i (1, 2, or 3). If $c_i = 1$, the line additionally has an integer x.

In other words, each query is in one of the three formats below.

$$
1 \ \ x
$$

2

3

Output

Print q lines, where q is the number of queries with $c_i = 2$.
The j-th line $(1 \leq j \leq q)$ should contain the response for the j-th such query.

Sample 1

Input	Output
8 1 4 1 3 1 2 1 1 3 2 1 0 2	1 2

The i-th line below shows the contents of A after the i-th query is processed in Sample Input 1.

- (4)
- $(4, 3)$
- $(4, 3, 2)$
- $(4, 3, 2, 1)$
- $(1, 2, 3, 4)$
- $(2, 3, 4)$
- $(2, 3, 4, 0)$
- $(3, 4, 0)$

Sample 2

Input	Output
9 1 5 1 5 1 3 2 3 2 1 6 3 2	5 3 5

The i-th line below shows the contents of A after the i-th query is processed in Sample Input 2.

- (5)
- $(5, 5)$
- $(5, 5, 3)$
- $(5, 3)$
- $(3, 5)$
- (5)
- $(5, 6)$
- $(5, 6)$
- (6)

Problem J. Berpizza

Time Limit 5000 ms
Mem Limit 524288 kB

Monocarp and Polycarp are working as waiters in Berpizza, a pizzeria located near the center of Bertown. Since they are waiters, their job is to serve the customers, but they choose whom they serve first differently.

At the start of the working day, there are no customers at the Berpizza. They come there one by one. When a customer comes into the pizzeria, she sits and waits for Monocarp or Polycarp to serve her. Monocarp has been working in Berpizza for just two weeks, so whenever he serves a customer, he simply chooses the one who came to Berpizza first, and serves that customer.

On the other hand, Polycarp is an experienced waiter at Berpizza, and he knows which customers are going to spend a lot of money at the pizzeria (and which aren't) as soon as he sees them. For each customer, Polycarp estimates the amount of money this customer can spend, and when he serves a customer, he chooses the one that is expected to leave the most money at Berpizza (in case there are several such customers, he chooses the one who came first among them).

Obviously, no customer can be served twice, so Monocarp and Polycarp choose which customer to serve only among those who haven't been served yet.

When the number of customers gets really high, it becomes difficult for both Monocarp and Polycarp to choose the customer they are going to serve. Your task is to write a program that makes these choices for them. Formally, your program should be able to process three types of queries:

- $1\ m$ — a customer comes to Berpizza, and Polycarp estimates the amount of money that they will spend as m;
- 2 — Monocarp serves a customer which came to the pizzeria first;
- 3 — Polycarp serves a customer which is expected to spend the largest amount of money at the pizzeria (if there are several such customers, the one that came to the pizzeria first is chosen).

For each query of types 2 and 3, report the number of the customer who was served (the customers are numbered in the order they come to the pizzeria, starting from 1).

Input

The first line contains one integer q ($2 \le q \le 5 \cdot 10^5$) — the number of queries.

Then q lines follow, each describing a query in one of the following formats:

- $1\ m$ ($1 \le m \le 5 \cdot 10^5$) — a customer comes to Berpizza, and Polycarp estimates the amount of money that they will spend as m;
- 2 — Monocarp serves a customer which came to the pizzeria first;
- 3 — Polycarp serves a customer which is expected to spend the largest amount of money at the pizzeria (if there are multiple such customers, the one that came to the pizzeria first is chosen).

Queries of type 2 and 3 are asked only when there exists at least one customer that hasn't been served yet. There is at least one query of type 2 or 3 in the input.

Output

For each query of type 2 or 3, print one integer — the number of the customer that has been served in that event. The customers are numbered in the order in which they come to the pizzeria, starting from 1.

Examples

Input	Output
8 1 8 1 10 1 6 3 2 1 9 2 3	2 1 3 4

Input	Output
6 1 8 1 10 1 8 3 3 3	2 1 3

Input	Output
8 1 103913 3 1 103913 1 103913 3 1 103913 1 103913 2	1 2 3