

Given is an ordered deck of  $n$  cards numbered 1 to  $n$  with card 1 at the top and card  $n$  at the bottom. The following operation is performed as long as there are at least two cards in the deck:

Throw away the top card and move the card that is now on the top of the deck to the bottom of the deck.

Your task is to find the sequence of discarded cards and the last, remaining card.

### Input

Each line of input (except the last) contains a number  $n \leq 50$ . The last line contains '0' and this line should not be processed.

### Output

For each number from the input produce two lines of output. The first line presents the sequence of discarded cards, the second line reports the last remaining card. No line will have leading or trailing spaces. See the sample for the expected format.

### Sample Input

```
7
19
10
6
0
```

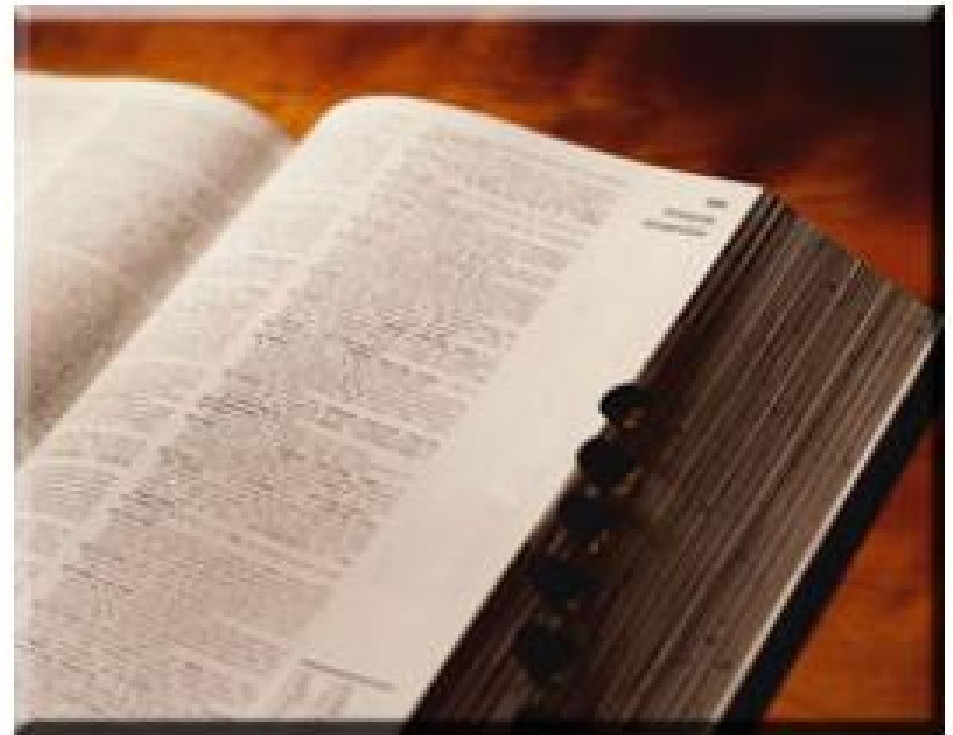
### Sample Output

```
Discarded cards: 1, 3, 5, 7, 4, 2
Remaining card: 6
Discarded cards: 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 4, 8, 12, 16, 2, 10, 18, 14
Remaining card: 6
Discarded cards: 1, 3, 5, 7, 9, 2, 6, 10, 8
Remaining card: 4
Discarded cards: 1, 3, 5, 2, 6
Remaining card: 4
```



Andy, 8, has a dream - he wants to produce his very own dictionary. This is not an easy task for him, as the number of words that he knows is, well, not quite enough. Instead of thinking up all the words himself, he has a brilliant idea. From his bookshelf he would pick one of his favourite story books, from which he would copy out all the distinct words. By arranging the words in alphabetical order, he is done! Of course, it is a really time-consuming job, and this is where a computer program is helpful.

You are asked to write a program that lists all the different words in the input text. In this problem, a word is defined as a consecutive sequence of alphabets, in upper and/or lower case. Words with only one letter are also to be considered. Furthermore, your program must be CaSe InSeNsItIvE. For example, words like “Apple”, “apple” or “APPLE” must be considered the same.



## Input

The input file is a text with no more than 5000 lines. An input line has at most 200 characters. Input is terminated by EOF.

## Output

Your output should give a list of different words that appears in the input text, one in a line. The words should all be in lower case, sorted in alphabetical order. You can be sure that the number of distinct words in the text does not exceed 5000.

Harry Potter is on a mission to destroy You-Know-Who's Horcruxes. The first Horcrux that he encountered in the Chamber of Secrets is Tom Riddle's diary. The diary was with Ginny and it forced her to open the Chamber of Secrets. Harry wants to know the different people who had ever possessed the diary to make sure they are not under its influence.

He has names of  $n$  people who possessed the diary in order. You need to tell, for each person, if he/she possessed the diary at some point before or not.

Formally, for a name  $s_i$  in the  $i$ -th line, output "YES" (without quotes) if there exists an index  $j$  such that  $s_i = s_j$  and  $j < i$ , otherwise, output "NO" (without quotes).

## Input

First line of input contains an integer  $n$  ( $1 \leq n \leq 100$ ) — the number of names in the list.

Next  $n$  lines each contain a string  $s_i$ , consisting of lowercase English letters. The length of each string is between 1 and 100.

## Output

Output  $n$  lines each containing either "YES" or "NO" (without quotes), depending on whether this string was already present in the stream or not.

You can print each letter in any case (upper or lower).

## Examples

input	Copy
6 tom lucius ginny harry ginny harry	
output	Copy
NO NO NO NO YES YES	