Codeforces Beta Round #59 (Div. 2)

A. Sinking Ship

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The ship crashed into a reef and is sinking. Now the entire crew must be evacuated. All n crew members have already lined up in a row (for convenience let's label them all from left to right with positive integers from 1 to n) and await further instructions. However, one should evacuate the crew properly, in a strict order. Specifically:

The first crew members to leave the ship are rats. Then women and children (both groups have the same priority) leave the ship. After that all men are evacuated from the ship. The captain leaves the sinking ship last.

If we cannot determine exactly who should leave the ship first for any two members of the crew by the rules from the previous paragraph, then the one who stands to the left in the line leaves the ship first (or in other words, the one whose number in the line is less).

For each crew member we know his status as a crew member, and also his name. All crew members have different names. Determine the order in which to evacuate the crew.

Input

The first line contains an integer n, which is the number of people in the crew ($1 \le n \le 100$). Then follow n lines. The i-th of those lines contains two words — the name of the crew member who is i-th in line, and his status on the ship. The words are separated by exactly one space. There are no other spaces in the line. The names consist of Latin letters, the first letter is uppercase, the rest are lowercase. The length of any name is from 1 to 10 characters. The status can have the following values: `rat` for a rat, `woman` for a woman, `child` for a child, `man` for a man, `captain` for the captain. The crew contains exactly one captain.

Output

Print n lines. The i-th of them should contain the name of the crew member who must be the i-th one to leave the ship.

Sample test(s)

input
6 Jack captain Alice woman Charlie man Teddy rat Bob child Julia woman

output
Teddy Alice Bob Julia Charlie Jack

B. Settlers' Training

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In a strategic computer game "Settlers II" one has to build defense structures to expand and protect the territory. Let's take one of these buildings. At the moment the defense structure accommodates exactly n soldiers. Within this task we can assume that the number of soldiers in the defense structure won't either increase or decrease.

Every soldier has a rank — some natural number from 1 to k. 1 stands for a private and k stands for a general. The higher the rank of the soldier is, the better he fights. Therefore, the player profits from having the soldiers of the highest possible rank.

To increase the ranks of soldiers they need to train. But the soldiers won't train for free, and each training session requires one golden coin. On each training session all the n soldiers are present.

At the end of each training session the soldiers' ranks increase as follows. First all the soldiers are divided into groups with the same rank, so that the least possible number of groups is formed. Then, within each of the groups where the soldiers below the rank k are present, exactly one soldier increases his rank by one.

You know the ranks of all n soldiers at the moment. Determine the number of golden coins that are needed to increase the ranks of all the soldiers to the rank k.

Input

The first line contains two integers n and k ($1 \le n, k \le 100$). They represent the number of soldiers and the number of different ranks correspondingly. The second line contains n numbers **in the non-decreasing order**. The i-th of them, a_i, represents the rank of the i-th soldier in the defense building ($1 \le i \le n$, $1 \le a_i \le k$).

Output

Print a single integer — the number of golden coins needed to raise all the soldiers to the maximal rank.

Sample test(s)

input
4 4 1 2 2 3
output
4

input
4 3 1 1 1 1
output
5

Note

In the first example the ranks will be raised in the following manner:

1 2 2 3 → 2 2 3 4 → 2 3 4 4 → 3 4 4 4 → 4 4 4 4

Thus totals to 4 training sessions that require 4 golden coins.

C. Bulls and Cows

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The "Bulls and Cows" game needs two people to play. The thinker thinks of a number and the guesser tries to guess it.

The thinker thinks of a four-digit number in the decimal system. All the digits in the number are different and the number may have a leading zero. It can't have more than one leading zero, because all it's digits should be different. The guesser tries to guess the number. He makes a series of guesses, trying experimental numbers and receives answers from the first person in the format "x bulls y cows". x represents the number of digits in the experimental number that occupy the same positions as in the sought number. y represents the number of digits of the experimental number that present in the sought number, but occupy different positions. Naturally, the experimental numbers, as well as the sought number, are represented by four-digit numbers where all digits are different and a leading zero can be present.

For example, let's suppose that the thinker thought of the number `0123`. Then the guessers' experimental number `1263` will receive a reply "1 bull 2 cows" (3 occupies the same positions in both numbers and 1 and 2 are present in both numbers but they occupy different positions). Also, the answer to number `8103` will be "2 bulls 1 cow" (analogically, 1 and 3 occupy the same positions and 0 occupies a different one).

When the guesser is answered "4 bulls 0 cows", the game is over.

Now the guesser has already made several guesses and wants to know whether his next guess can possibly be the last one.

Input

The first input line contains an integer n ($1 \le n \le 10$) which represents the number of already made guesses. Then follow n lines in the form of "a_i b_i c_i", where a_i is the i-th experimental number, b_i is the number of bulls, c_i is the number of cows ($1 \le i \le n$, $0 \le b_i, c_i, b_i + c_i \le 4$). The experimental numbers are correct, i.e., each of them contains exactly four digits, in each of them all the four digits are different, and there can be a leading zero. All the experimental numbers are different. As the guesser hasn't guessed the number yet, the answer "4 bulls 0 cows" is not present.

Output

If the input data is enough to determine the sought number, print the number with four digits on a single line. If it has less than four digits, add leading zero. If the data is not enough, print "`Need more data`" without the quotes. If the thinker happens to have made a mistake in his replies, print "`Incorrect data`" without the quotes.

Sample test(s)

input
2 1263 1 2 8103 2 1
output
Need more data

input
2 1234 2 2 1256 0 2
output
2134

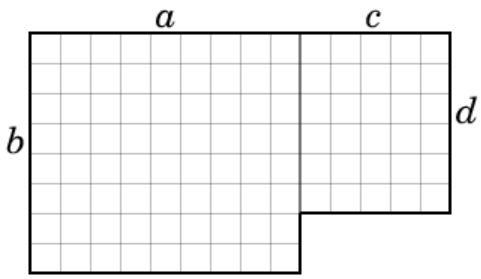
input
2 0123 1 1 4567 1 2
output
Incorrect data

D. Dividing Island

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A revolution took place on the Buka Island. New government replaced the old one. The new government includes n parties and each of them is entitled to some part of the island according to their contribution to the revolution. However, they can't divide the island.

The island can be conventionally represented as two rectangles $a \times b$ and $c \times d$ unit squares in size correspondingly. The rectangles are located close to each other. At that, one of the sides with the length of a and one of the sides with the length of c lie on one line. You can see this in more details on the picture.



The i-th party is entitled to a part of the island equal to x_i unit squares. Every such part should fully cover several squares of the island (it is not allowed to cover the squares partially) and be a connected figure. A "connected figure" presupposes that from any square of this party one can move to any other square of the same party moving through edge-adjacent squares also belonging to that party.

Your task is to divide the island between parties.

Input

The first line contains 5 space-separated integers — a, b, c, d and n ($1 \le a, b, c, d \le 50, b \ne d, 1 \le n \le 26$). The second line contains n space-separated numbers. The i-th of them is equal to number x_i ($1 \le x_i \le a \times b + c \times d$). It is guaranteed that $\sum_{i=1}^{n} x_i = a \times b + c \times d$.

Output

If dividing the island between parties in the required manner is impossible, print "`NO`" (without the quotes). Otherwise, print "`YES`" (also without the quotes) and, starting from the next line, print $max(b, d)$ lines each containing $a + c$ characters. To mark what square should belong to what party, use lowercase Latin letters. For the party that is first in order in the input data, use "`a`", for the second one use "`b`" and so on. Use "`.`" for the squares that belong to the sea. The first symbol of the second line of the output data should correspond to the square that belongs to the rectangle $a \times b$. The last symbol of the second line should correspond to the square that belongs to the rectangle $c \times d$.

If there are several solutions output any.

Sample test(s)

input
3 4 2 2 3 5 8 3
output
YES aaabb aabbb cbb.. ccb..

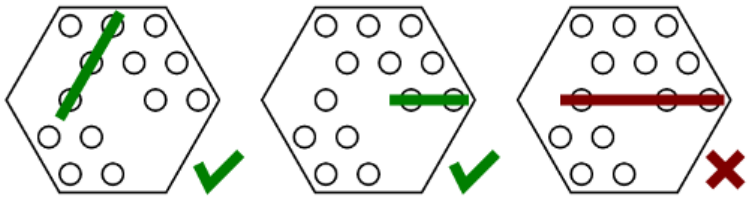
input
3 2 1 4 4 1 2 3 4
output
YES abbd cccd ...d ...d

E. Sweets Game

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Karlsson has visited Lillebror again. They found a box of chocolates and a big whipped cream cake at Lillebror's place. Karlsson immediately suggested to divide the sweets fairly between Lillebror and himself. Specifically, to play together a game he has just invented with the chocolates. The winner will get the cake as a reward.

The box of chocolates has the form of a hexagon. It contains 19 cells for the chocolates, some of which contain a chocolate. The players move in turns. During one move it is allowed to eat one or several chocolates that lay in the neighboring cells on one line, parallel to one of the box's sides. The picture below shows the examples of allowed moves and of an unacceptable one. The player who cannot make a move loses.



Karlsson makes the first move as he is Lillebror's guest and not vice versa. The players play optimally. Determine who will get the cake.

Input

The input data contains 5 lines, containing 19 words consisting of one symbol. The word "`O`" means that the cell contains a chocolate and a "`.`" stands for an empty cell. It is guaranteed that the box contains at least one chocolate. See the examples for better understanding.

Output

If Karlsson gets the cake, print "`Karlsson`" (without the quotes), otherwise print "`Lillebror`" (yet again without the quotes).

Sample test(s)

input

```
  . . .
 . . O .
. . O O .
 . . . .
  . . .
```

output

```
Lillebror
```

input

```
  . . .
 . . . O
. . . O .
 O . O .
  . O .
```

output

```
Karlsson
```

Codeforces (c) Copyright 2010-2015 Mike Mirzayanov
The only programming contests Web 2.0 platform