Technocup 2017 - Elimination Round 1 (Unofficially Open for Everyone, Rated for Div. 2)

A. Transformation: from A to B

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasily has a number a, which he wants to turn into a number b. For this purpose, he can do two types of operations:

- multiply the current number by 2 (that is, replace the number x by $2 \cdot x$);
- append the digit 1 to the right of current number (that is, replace the number x by $10 \cdot x + 1$).

You need to help Vasily to transform the number a into the number b using only the operations described above, or find that it is impossible.

Note that in this task you are not required to minimize the number of operations. It suffices to find any way to transform a into b.

Input

The first line contains two positive integers a and b ($1 \le a < b \le 10^9$) — the number which Vasily has and the number he wants to have.

Output

If there is no way to get b from a, print "NO" (without quotes).

Otherwise print three lines. On the first line print "YES" (without quotes). The second line should contain single integer k — the length of the transformation sequence. On the third line print the sequence of transformations $x_1, x_2, ..., x_k$, where:

- x_1 should be equal to a,
- x_k should be equal to b,
- x_i should be obtained from x_{i-1} using any of two described operations ($1 < i \le k$).

If there are multiple answers, print any of them.

Examples

input
2 162
output
YES 5 2 4 8 81 162

input
4 42
output
NO

input
100 40021
output
YES 5 100 200 2001 4002 40021

B. Bill Total Value

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasily exited from a store and now he wants to recheck the total price of all purchases in his bill. The bill is a string in which the names of the purchases and their prices are printed in a row without any spaces. Check has the format "$name_1 price_1 name_2 price_2 ... name_n price_n$", where $name_i$ (name of the i-th purchase) is a non-empty string of length not more than 10, consisting of lowercase English letters, and $price_i$ (the price of the i-th purchase) is a non-empty string, consisting of digits and dots (decimal points). It is possible that purchases with equal names have different prices.

The price of each purchase is written in the following format. If the price is an integer number of dollars then cents are not written.

Otherwise, after the number of dollars a dot (decimal point) is written followed by cents **in a two-digit format** (if number of cents is between 1 and 9 inclusively, there is a leading zero).

Also, every three digits (from less significant to the most) in dollars are separated by dot (decimal point). No extra leading zeroes are allowed. The price always starts with a digit and ends with a digit.

For example:

- "234", "1.544", "149.431.10", "0.99" and "123.05" are valid prices,
- ".333", "3.33.11", "12.00", ".33", "0.1234" and "1.2" are not valid.

Write a program that will find the total price of all purchases in the given bill.

Input

The only line of the input contains a non-empty string s with length not greater than 1000 — the content of the bill.

It is guaranteed that the bill meets the format described above. It is guaranteed that each price in the bill is not less than one cent and not greater than 10^6 dollars.

Output

Print the total price **exactly in the same format** as prices given in the input.

Examples

input
chipsy48.32televizor12.390
output
12.438.32

input
a1b2c3.38
output
6.38

input
aa0.01t0.03
output
0.04

C. Guess the Array

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is an interactive problem. You should use `flush` *operation after each printed line. For example, in C++ you should use* `fflush(stdout)`*, in Java you should use* `System.out.flush()`*, and in Pascal —* `flush(output)`*.*

In this problem you should guess an array a which is unknown for you. The only information you have initially is the length n of the array a.

The only allowed action is to ask the sum of two elements by their indices. Formally, you can print two indices i and j (the indices should be **distinct**). Then your program should read the response: the single integer equals to $a_i + a_j$.

It is easy to prove that it is always possible to guess the array using at most n requests.

Write a program that will guess the array a by making at most n requests.

Interaction

In each test your program should guess a single array.

The input starts with a line containing integer n ($3 \le n \le 5000$) — the length of the array. Your program should read it at first.

After that your program should print to the standard output the requests about the sum of two elements or inform that the array is guessed.

- In case your program is making a request to ask the sum of two elements, it should print line in the format "`? i j`" (i and j are distinct integers between 1 and n), where i and j are indices in the array a.
- In case your program informs that the array is guessed, it should print line in the format "$!\ a_1\ a_2\ \ldots\ a_n$" (it is guaranteed that all a_i are positive integers not exceeding 10^5), where a_i is the i-th element of the array a.

The response on a request is a single integer equal to $a_i + a_j$, printed on a separate line.

Your program can do at most n requests. Note that the final line «$!\ a_1\ a_2\ \ldots\ a_n$» is not counted as a request.

Do not forget about `flush` operation after each printed line.

After you program prints the guessed array, it should terminate normally.

Example

input
5 9 7 9 11 6

output
? 1 5 ? 2 3 ? 4 1 ? 5 2 ? 3 4 ! 4 6 1 5 5

Note

The format of a test to make a hack is:

- The first line contains an integer number n ($3 \le n \le 5000$) — the length of the array.
- The second line contains n numbers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$) — the elements of the array to guess.

D. T-shirts Distribution

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The organizers of a programming contest have decided to present t-shirts to participants. There are six different t-shirts sizes in this problem: `S`, `M`, `L`, `XL`, `XXL`, `XXXL` (sizes are listed in increasing order). The t-shirts are already prepared. For each size from `S` to `XXXL` you are given the number of t-shirts of this size.

During the registration, the organizers asked each of the n participants about the t-shirt size he wants. If a participant hesitated between two sizes, he could specify two neighboring sizes — this means that any of these two sizes suits him.

Write a program that will determine whether it is possible to present a t-shirt to each participant of the competition, or not. Of course, each participant should get a t-shirt of proper size:

- the size he wanted, if he specified one size;
- any of the two neibouring sizes, if he specified two sizes.

If it is possible, the program should find any valid distribution of the t-shirts.

Input

The first line of the input contains six non-negative integers — the number of t-shirts of each size. The numbers are given for the sizes `S`, `M`, `L`, `XL`, `XXL`, `XXXL`, respectively. The total number of t-shirts doesn't exceed $100\,000$.

The second line contains positive integer n ($1 \le n \le 100\,000$) — the number of participants.

The following n lines contain the sizes specified by the participants, one line per participant. The i-th line contains information provided by the i-th participant: single size or two sizes separated by comma (without any spaces). If there are two sizes, the sizes are written in increasing order. It is guaranteed that two sizes separated by comma are neighboring.

Output

If it is not possible to present a t-shirt to each participant, print «`NO`» (without quotes).

Otherwise, print $n + 1$ lines. In the first line print «`YES`» (without quotes). In the following n lines print the t-shirt sizes the orginizers should give to participants, one per line. The order of the participants should be the same as in the input.

If there are multiple solutions, print any of them.

Examples

input
0 1 0 1 1 0 3 XL S,M XL,XXL
output
YES XL M XXL

input
1 1 2 0 1 1 5 S M S,M XXL,XXXL XL,XXL
output
NO

E. Games on a CD

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Several years ago Tolya had n computer games and at some point of time he decided to burn them to CD. After that he wrote down the names of the games one after another in a circle on the CD **in clockwise order**. The names were distinct, the length of each name was equal to k. The names didn't overlap.

Thus, there is a cyclic string of length $n \cdot k$ written on the CD.

Several years have passed and now Tolya can't remember which games he burned to his CD. He knows that there were g popular games that days. All of the games he burned were among these g games, and **no game was burned more than once**.

You have to restore any valid list of games Tolya could burn to the CD several years ago.

Input

The first line of the input contains two positive integers n and k ($1 \le n \le 10^5$, $1 \le k \le 10^5$) — the amount of games Tolya burned to the CD, and the length of each of the names.

The second line of the input contains one string consisting of lowercase English letters — the string Tolya wrote on the CD, split in arbitrary place. The length of the string is $n \cdot k$. It is guaranteed that the length is not greater than 10^6.

The third line of the input contains one positive integer g ($n \le g \le 10^5$) — the amount of popular games that could be written on the CD. It is guaranteed that the total length of names of all popular games is not greater than $2 \cdot 10^6$.

Each of the next g lines contains a single string — the name of some popular game. Each name consists of lowercase English letters and has length k. It is guaranteed that the names are distinct.

Output

If there is no answer, print "NO" (without quotes).

Otherwise, print two lines. In the first line print "YES" (without quotes). In the second line, print n integers — the games which names were written on the CD. You should print games in the order they could have been written on the CD, it means, **in clockwise order**. You can print games starting from any position. Remember, that no game was burned to the CD more than once. If there are several possible answers, print any of them.

Examples

input
3 1 abc 4 b a c d
output
YES 2 1 3

input
4 2 aabbccdd 4 dd ab bc cd
output
NO

F. Polycarp's problems

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

Polycarp is an experienced participant in Codehorses programming contests. Now he wants to become a problemsetter.

He sent to the coordinator a set of n problems. Each problem has it's quality, the quality of the i-th problem is a_i (a_i can be positive, negative or equal to zero). The problems are ordered by expected difficulty, but the difficulty is not related to the quality in any way. The easiest problem has index 1, the hardest problem has index n.

The coordinator's mood is equal to q now. After reading a problem, the mood changes by it's quality. It means that after the coordinator reads a problem with quality b, the value b is added to his mood. The coordinator always reads problems one by one from the easiest to the hardest, it's impossible to change the order of the problems.

If after reading some problem the coordinator's mood becomes negative, he immediately stops reading and rejects the problemset.

Polycarp wants to remove the minimum number of problems from his problemset to make the coordinator's mood non-negative at any moment of time. Polycarp is not sure about the current coordinator's mood, but he has m guesses "the current coordinator's mood $q = b_i$".

For each of m guesses, find the minimum number of problems Polycarp needs to remove so that the coordinator's mood will always be greater or equal to 0 while he reads problems from the easiest of the remaining problems to the hardest.

Input

The first line of input contains two integers n and m ($1 \le n \le 750$, $1 \le m \le 200\,000$) — the number of problems in the problemset and the number of guesses about the current coordinator's mood.

The second line of input contains n integers $a_1, a_2, ..., a_n$ ($-10^9 \le a_i \le 10^9$) — the qualities of the problems in order of increasing difficulty.

The third line of input contains m integers $b_1, b_2, ..., b_m$ ($0 \le b_i \le 10^{15}$) — the guesses of the current coordinator's mood q.

Output

Print m lines, in i-th line print single integer — the answer to the problem with $q = b_i$.

Example

input
6 3 8 -5 -4 1 -7 4 0 7 3
output
2 0 1

Codeforces (c) Copyright 2010-2017 Mike Mirzayanov
The only programming contests Web 2.0 platform