[DivC] Contest #2

A. Vladik and flights

2 seconds, 256 megabytes

Vladik is a competitive programmer. This year he is going to win the International Olympiad in Informatics. But it is not as easy as it sounds: the question Vladik face now is to find the cheapest way to get to the olympiad.

Vladik knows n airports. All the airports are located on a straight line. Each airport has unique id from 1 to n, Vladik's house is situated next to the airport with id a, and the place of the olympiad is situated next to the airport with id b. It is possible that Vladik's house and the place of the olympiad are located near the same airport.

To get to the olympiad, Vladik can fly between any pair of airports any number of times, but he has to start his route at the airport a and finish it at the airport b.

Each airport belongs to one of two companies. The cost of flight from the airport i to the airport j is zero if both airports belong to the same company, and $|i - j|$ if they belong to different companies.

Print the minimum cost Vladik has to pay to get to the olympiad.

Input

The first line contains three integers n, a, and b ($1 \le n \le 10^5$, $1 \le a, b \le n$) — the number of airports, the id of the airport from which Vladik starts his route and the id of the airport which he has to reach.

The second line contains a string with length n, which consists only of characters 0 and 1. If the i-th character in this string is 0, then i-th airport belongs to first company, otherwise it belongs to the second.

Output

Print single integer — the minimum cost Vladik has to pay to get to the olympiad.

input
```
4 1 4
1010
```
output
```
1
```

input
```
5 5 2
10110
```
output
```
0
```

In the first example Vladik can fly to the airport 2 at first and pay $|1 - 2| = 1$ (because the airports belong to different companies), and then fly from the airport 2 to the airport 4 for free (because the airports belong to the same company). So the cost of the whole flight is equal to 1. It's impossible to get to the olympiad for free, so the answer is equal to 1.

In the second example Vladik can fly directly from the airport 5 to the airport 2, because they belong to the same company.

B. AppendAppendAppend

1 second, 256 megabytes

Momo has a string s which he plays with. After each day, he takes the initial string and appends it to the one that he currently has, that is, if the string is "abc", on the first day the string will remain "abc", on the second day the string will become "$abcabc$", on the third day the string becomes "$abcabcabc$", and so on.

Bobo has another string t. Bobo is curious whether his string t can be found as a subsequence in Momo's string. He is curious what is the first day when this property will hold.

A string a is a subsequence of a string b if a can be obtained from b by deletion of several (possibly, zero or all) elements.

Input

The first line of the input contains Momo's string s.

The second line of the input contains Bobo's string t.

Both strings have lengths at least 1 and at most $5 \cdot 10^5$ and contain only lowercase letters of the English alphabet.

Output

Output a single integer — the number of the day when Bobo's string will be a subsequence of Momo's string for the first time.

It is guaranteed that for all test cases in this problem such a day exists.

input
```
dwalkcake
cakewalk
```
output
```
2
```

During the first day, the string is "$dwalkcake$", and doesn't contain "$cakewalk$" as a subsequence.

During the second day, the string is "$dwalkcakedwalkcake$", and it contains "$cakewalk$" as a subsequence.

C. Guess the Array

1 second, 256 megabytes

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.

input
```
5

9

7

9

11

6
```

output
```

? 1 5

? 2 3

? 4 1

? 5 2

? 3 4

! 4 6 1 5 5
```

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. Chloe and the sequence

1 second, 256 megabytes

Chloe, the same as Vladik, is a competitive programmer. She didn't have any problems to get to the olympiad like Vladik, but she was confused by the task proposed on the olympiad.

Let's consider the following algorithm of generating a sequence of integers. Initially we have a sequence consisting of a single element equal to 1. Then we perform $(n - 1)$ steps. On each step we take the sequence we've got on the previous step, append it to the end of itself and insert in the middle the minimum positive integer we haven't used before. For example, we get the sequence $[1, 2, 1]$ after the first step, the sequence $[1, 2, 1, 3, 1, 2, 1]$ after the second step.

The task is to find the value of the element with index k (the elements are numbered from 1) in the obtained sequence, i. e. after $(n - 1)$ steps.

Please help Chloe to solve the problem!

Input
The only line contains two integers n and k $(1 \le n \le 50, 1 \le k \le 2^n - 1)$.

Output
Print single integer — the integer at the k-th position in the obtained sequence.

input
```
3 2
```

output
```
2
```

input
```
4 8
```

output
```
4
```

In the first sample the obtained sequence is $[1, 2, 1, 3, 1, 2, 1]$. The number on the second position is 2.

In the second sample the obtained sequence is $[1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1]$. The number on the eighth position is 4.

E. K-th Not Divisible by n

1 second, 256 megabytes

You are given two positive integers n and k. Print the k-th positive integer that is not divisible by n.

For example, if $n = 3$, and $k = 7$, then all numbers that are not divisible by 3 are: $1, 2, 4, 5, 7, 8, 10, 11, 13 \ldots$. The 7-th number among them is 10.

Input
The first line contains an integer t $(1 \le t \le 1000)$ — the number of test cases in the input. Next, t test cases are given, one per line.

Each test case is two positive integers n $(2 \le n \le 10^9)$ and k $(1 \le k \le 10^9)$.

Output
For each test case print the k-th positive integer that is not divisible by n.

input
```
6
3 7
4 12
2 1000000000
7 97
1000000000 1000000000
2 1
```

output
```
10
15
1999999999
113
1000000001
1
```

F. Yet Another String Game

2 seconds, 512 megabytes

Homer has two friends Alice and Bob. Both of them are string fans.

One day, Alice and Bob decide to play a game on a string $s = s_1 s_2 \ldots s_n$ of length n consisting of lowercase English letters. They move in turns alternatively and **Alice makes the first move**.

In a move, a player **must** choose an index i $(1 \le i \le n)$ that has not been chosen before, and change s_i to any other lowercase English letter c that $c \ne s_i$.

When all indices have been chosen, the game ends.

The goal of Alice is to make the final string lexicographically as small as possible, while the goal of Bob is to make the final string lexicographically as large as possible. Both of them are game experts, so they always play games optimally. Homer is not a game expert, so he wonders what the final string will be.

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a \ne b$;
- in the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Input
Each test contains multiple test cases. The first line contains t $(1 \le t \le 1000)$ — the number of test cases. Description of the test cases follows.

The only line of each test case contains a single string s $(1 \le |s| \le 50)$ consisting of lowercase English letters.

Output
For each test case, print the final string in a single line.

input

```
3
a
bbbb
az
```

output

```
b
azaz
by
```

In the first test case: Alice makes the first move and must change the only letter to a different one, so she changes it to 'b'.

In the second test case: Alice changes the first letter to 'a', then Bob changes the second letter to 'z', Alice changes the third letter to 'a' and then Bob changes the fourth letter to 'z'.

In the third test case: Alice changes the first letter to 'b', and then Bob changes the second letter to 'y'.

G. Watto and Mechanism

3 seconds, 256 megabytes

Watto, the owner of a spare parts store, has recently got an order for the mechanism that can process strings in a certain way. Initially the memory of the mechanism is filled with n strings. Then the mechanism should be able to process queries of the following type: "Given string s, determine if the memory of the mechanism contains string t that consists of the same number of characters as s and differs from s in exactly one position".

Watto has already compiled the mechanism, all that's left is to write a program for it and check it on the data consisting of n initial lines and m queries. He decided to entrust this job to you.

Input

The first line contains two non-negative numbers n and m ($0 \le n \le 3 \cdot 10^5$, $0 \le m \le 3 \cdot 10^5$) — the number of the initial strings and the number of queries, respectively.

Next follow n non-empty strings that are uploaded to the memory of the mechanism.

Next follow m non-empty strings that are the queries to the mechanism.

The total length of lines in the input doesn't exceed $6 \cdot 10^5$. Each line consists **only** of letters 'a', 'b', 'c'.

Output

For each query print on a single line "YES" (without the quotes), if the memory of the mechanism contains the required string, otherwise print "NO" (without the quotes).

input

```
2 3
aaaaa
acacaca
aabaa
ccacacc
caaac
```

output

```
YES
NO
NO
```

H. Weird Sum

2 seconds, 256 megabytes

Egor has a table of size $n \times m$, with lines numbered from 1 to n and columns numbered from 1 to m. Each cell has a color that can be presented as an integer from 1 to 10^5.

Let us denote the cell that lies in the intersection of the r-th row and the c-th column as (r, c). We define the manhattan distance between two cells (r_1, c_1) and (r_2, c_2) as the length of a shortest path between them where each consecutive cells in the path must have a common side. The path can go through cells of any color. For example, in the table 3×4 the manhattan distance between $(1, 2)$ and $(3, 3)$ is 3, one of the shortest paths is the following: $(1, 2) \to (2, 2) \to (2, 3) \to (3, 3)$.

Egor decided to calculate the sum of manhattan distances between each pair of cells of the same color. Help him to calculate this sum.

Input

The first line contains two integers n and m ($1 \le n \le m$, $n \cdot m \le 100\,000$) — number of rows and columns in the table.

Each of next n lines describes a row of the table. The i-th line contains m integers $c_{i1}, c_{i2}, \ldots, c_{im}$ ($1 \le c_{ij} \le 100\,000$) — colors of cells in the i-th row.

Output

Print one integer — the the sum of manhattan distances between each pair of cells of the same color.

input

```
2 3
1 2 3
3 2 1
```

output

```
7
```

input

```
3 4
1 1 2 2
2 1 1 2
2 2 1 1
```

output

```
76
```

input

```
4 4
1 1 2 3
2 1 1 2
3 1 2 1
1 1 2 1
```

output

```
129
```

In the first sample there are three pairs of cells of same color: in cells $(1, 1)$ and $(2, 3)$, in cells $(1, 2)$ and $(2, 2)$, in cells $(1, 3)$ and $(2, 1)$. The manhattan distances between them are 3, 1 and 3, the sum is 7.

I. Chips Moving

1 second, 256 megabytes

You are given n chips on a number line. The i-th chip is placed at the integer coordinate x_i. Some chips **can have equal coordinates**.

You can perform each of the two following types of moves any (possibly, zero) number of times on any chip:

- Move the chip i by 2 to the left or 2 to the right **for free** (i.e. replace the current coordinate x_i with $x_i - 2$ or with $x_i + 2$);
- move the chip i by 1 to the left or 1 to the right and pay **one coin** for this move (i.e. replace the current coordinate x_i with $x_i - 1$ or with $x_i + 1$).

Note that it's allowed to move chips to any integer coordinate, including negative and zero.

Your task is to find the minimum total number of coins required to move all n chips to the same coordinate (i.e. all x_i should be equal after some sequence of moves).

Input

The first line of the input contains one integer n ($1 \le n \le 100$) — the number of chips.

The second line of the input contains n integers $x_1, x_2, \dots, x_n$ ($1 \le x_i \le 10^9$), where x_i is the coordinate of the i-th chip.

Output

Print one integer — the minimum total number of coins required to move all n chips to the same coordinate.

input
```
3
1 2 3
```
output
```
1
```

input
```
5
2 2 2 3 3
```
output
```
2
```

In the first example you need to move the first chip by 2 to the right and the second chip by 1 to the right or move the third chip by 2 to the left and the second chip by 1 to the left so the answer is 1.

In the second example you need to move two chips with coordinate 3 by 1 to the left so the answer is 2.

J. Little Xor

2 seconds, 256 megabytes

Little Petya likes arrays that consist of non-negative integers a lot. Recently his mom has presented him one such array consisting of n elements. Petya immediately decided to find there a segment of consecutive elements, such that the xor of all numbers from this segment was maximal possible. Help him with that.

The xor operation is the bitwise exclusive "`OR`", that is denoted as "`xor`" in Pascal and "`^`" in C/C++/Java.

Input

The first line contains integer n ($1 \le n \le 100$) — the number of elements in the array. The second line contains the space-separated integers from the array. All numbers are non-negative integers strictly less than 2^{30}.

Output

Print a single integer — the required maximal xor of a segment of consecutive elements.

input
```
5
1 2 1 1 2
```
output
```
3
```

input
```
3
1 2 7
```
output
```
7
```

input
```
4
4 2 4 8
```
output
```
14
```

In the first sample one of the optimal segments is the segment that consists of the first and the second array elements, if we consider the array elements indexed starting from one.

The second sample contains only one optimal segment, which contains exactly one array element (element with index three).

K. Sum of Round Numbers

1 second, 256 megabytes

A positive (strictly greater than zero) integer is called *round* if it is of the form `d00...0`. In other words, a positive integer is round if all its digits except the leftmost (most significant) are equal to zero. In particular, all numbers from 1 to 9 (inclusive) are round.

For example, the following numbers are round: $4000, 1, 9, 800, 90$. The following numbers are **not** round: $110, 707, 222, 1001$.

You are given a positive integer n ($1 \le n \le 10^4$). Represent the number n as a sum of round numbers using the minimum number of summands (addends). In other words, you need to represent the given number n as a sum of the least number of terms, each of which is a round number.

Input

The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases in the input. Then t test cases follow.

Each test case is a line containing an integer n ($1 \le n \le 10^4$).

Output

Print t answers to the test cases. Each answer must begin with an integer k — the minimum number of summands. Next, k terms must follow, each of which is a round number, and their sum is n. The terms can be printed in any order. If there are several answers, print any of them.

input
```
5
5009
7
9876
10000
10
```
output
```
2
5000 9
1
7
4
800 70 6 9000
1
10000
1
10
```

L. Lisa and the Martians

3 seconds, 512 megabytes

Lisa was kidnapped by martians! It okay, because she has watched a lot of TV shows about aliens, so she knows what awaits her. Let's call integer *martian* if it is **a non-negative integer** and **strictly less than** 2^k, for example, when $k = 12$, the numbers $51, 1960, 0$ are *martian*, and the numbers $\pi, -1, \frac{21}{8}, 4096$ are not.

The aliens will give Lisa n *martian* numbers $a_1, a_2, \dots, a_n$. Then they will ask her to name any *martian* number x. After that, Lisa will select a pair of numbers a_i, a_j ($i \ne j$) in the given sequence and count $(a_i \oplus x) \& (a_j \oplus x)$. The operation $\oplus$ means Bitwise exclusive OR, the operation $\&$ means Bitwise And. For example,
$(5 \oplus 17) \& (23 \oplus 17) = (00101_2 \oplus 10001_2) \& (10111_2 \oplus 10001_2) = 10100_2 \& 0011$
.

Lisa is sure that the higher the calculated value, the higher her chances of returning home. Help the girl choose such i, j, x that maximize the calculated value.

Input

The first line contains an integer t ($1 \le t \le 10^4$) — number of testcases.

Each testcase is described by two lines.

The first line contains integers n, k ($2 \le n \le 2 \cdot 10^5$, $1 \le k \le 30$) — the length of the sequence of *martian* numbers and the value of k.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i < 2^k$) — a sequence of *martian* numbers.

It is guaranteed that the sum of n over all testcases **does not exceed** $2 \cdot 10^5$.

Output

For each testcase, print three integers i, j, x ($1 \le i, j \le n$, $i \ne j$, $0 \le x < 2^k$). The value of $(a_i \oplus x) \& (a_j \oplus x)$ should be the maximum possible.

If there are several solutions, you can print any one.

input
```
10
5 4
3 9 1 4 13
3 1
1 0 1
6 12
144 1580 1024 100 9 13
4 3
7 3 0 4
3 2
0 0 1
2 4
12 2
9 4
6 14 9 4 4 4 5 10 2
2 1
1 0
2 4
11 4
9 4
2 11 10 1 6 9 11 0 5
```
output
```
1 3 14
1 3 0
5 6 4082
2 3 7
1 2 3
1 2 15
4 5 11
1 2 0
1 2 0
2 7 4
```

First testcase: $(3 \oplus 14)\&(1 \oplus 14) = (0011_2 \oplus 1110_2)\&(0001_2 \oplus 1110_2) = 1101_2 = 1101_2\&1111_2 = 1101_2 = 13$.

Second testcase: $(1 \oplus 0)\&(1 \oplus 0) = 1$.

Third testcase: $(9 \oplus 4082)\&(13 \oplus 4082) = 4091$.

Fourth testcase: $(3 \oplus 7)\&(0 \oplus 7) = 4$.

M. Set of Strings

1 second, 256 megabytes

You are given a string q. A sequence of k strings $s_1, s_2, \ldots, s_k$ is called *beautiful*, if the concatenation of these strings is string q (formally, $s_1 + s_2 + \ldots + s_k = q$) and the first characters of these strings are distinct.

Find any *beautiful* sequence of strings or determine that the *beautiful* sequence doesn't exist.

Input

The first line contains a positive integer k ($1 \le k \le 26$) — the number of strings that should be in a *beautiful* sequence.

The second line contains string q, consisting of lowercase Latin letters. The length of the string is within range from 1 to 100, inclusive.

Output

If such sequence doesn't exist, then print in a single line "NO" (without the quotes). Otherwise, print in the first line "YES" (without the quotes) and in the next k lines print the *beautiful* sequence of strings $s_1, s_2, \ldots, s_k$.

If there are multiple possible answers, print any of them.

input
```
1
abca
```
output
```
YES
abca
```

input
```
2
aaacas
```
output
```
YES
aaa
cas
```

input
```
4
abc
```
output
```
NO
```

In the second sample there are two possible answers: $\{"aaaca", "s"\}$ and $\{"aaa", "cas"\}$.