GPC-UPC Brute Force I Contest

A. Far Relative’s Problem

2 seconds, 256 megabytes

Famil Door wants to celebrate his birthday with his friends from Far Far Away. He has n friends and each of them can come to the party in a specific range of days of the year from a_i to b_i. Of course, Famil Door wants to have as many friends celebrating together with him as possible.

Far cars are as weird as Far Far Away citizens, so they can only carry two people of opposite gender, that is exactly one male and one female. However, Far is so far from here that no other transportation may be used to get to the party.

Famil Door should select some day of the year and invite some of his friends, such that they all are available at this moment and the number of male friends invited is equal to the number of female friends invited. Find the maximum number of friends that may present at the party.

Input

The first line of the input contains a single integer n $(1 \le n \le 5000)$ — then number of Famil Door's friends.

Then follow n lines, that describe the friends. Each line starts with a capital letter 'F' for female friends and with a capital letter 'M' for male friends. Then follow two integers a_i and b_i $(1 \le a_i \le b_i \le 366)$, providing that the i-th friend can come to the party from day a_i to day b_i inclusive.

Output

Print the maximum number of people that may come to Famil Door's party.

input
```
4
M 151 307
F 343 352
F 117 145
M 24 128
```
output
```
2
```

input
```
6
M 128 130
F 128 131
F 131 140
F 131 141
M 131 200
M 140 200
```
output
```
4
```

In the first sample, friends 3 and 4 can come on any day in range $[117, 128]$.

In the second sample, friends with indices 3, 4, 5 and 6 can come on day 140.

B. Restoring Painting

1 second, 256 megabytes

Vasya works as a watchman in the gallery. Unfortunately, one of the most expensive paintings was stolen while he was on duty. He doesn't want to be fired, so he has to quickly restore the painting. He remembers some facts about it.

- The painting is a square 3×3, each cell contains a single integer from 1 to n, and different cells may contain either different or equal integers.
- The sum of integers in each of four squares 2×2 is equal to the sum of integers in the top left square 2×2.
- Four elements a, b, c and d are known and are located as shown on the picture below.

?	a	?
b	?	c
?	d	?

Help Vasya find out the number of distinct squares the satisfy all the conditions above. Note, that this number may be equal to 0, meaning Vasya remembers something wrong.

Two squares are considered to be different, if there exists a cell that contains two different integers in different squares.

Input

The first line of the input contains five integers n, a, b, c and d $(1 \le n \le 100\,000, 1 \le a, b, c, d \le n)$ — maximum possible value of an integer in the cell and four integers that Vasya remembers.

Output

Print one integer — the number of distinct valid squares.

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

input
```
3 3 1 2 3
```
output
```
6
```

Below are all the possible paintings for the first sample.

2	1	2
1	1	1
1	2	1

2	1	2
1	2	1
1	2	1

In the second sample, only paintings displayed below satisfy all the rules.

2	3	1
1	1	2
2	3	1

2	3	1
1	2	2
2	3	1

2	3	1
1	3	2
2	3	1

3	3	2
1	1	2
3	3	2

3	3	2
1	2	2
3	3	2

3	3	2
1	3	2
3	3	2

C. Money Transfers

1 second, 256 megabytes

There are n banks in the city where Vasya lives, they are located in a circle, such that any two banks are neighbouring if their indices differ by no more than 1. Also, bank 1 and bank n are neighbours if $n > 1$. No bank is a neighbour of itself.

Vasya has an account in each bank. Its balance may be negative, meaning Vasya owes some money to this bank.

There is only one type of operations available: transfer some amount of money from any bank to account in any **neighbouring** bank. There are no restrictions on the size of the sum being transferred or balance requirements to perform this operation.

Vasya doesn't like to deal with large numbers, so he asks you to determine the minimum number of operations required to change the balance of each bank account to zero. It's guaranteed, that this is possible to achieve, that is, the total balance of Vasya in all banks is equal to zero.

Input

The first line of the input contains a single integer n $(1 \le n \le 100\,000)$ — the number of banks.

The second line contains n integers a_i $(-10^9 \le a_i \le 10^9)$, the i-th of them is equal to the initial balance of the account in the i-th bank. It's guaranteed that the sum of all a_i is equal to 0.

Output

Print the minimum number of operations required to change balance in each bank to zero.

input
```
3
5 0 -5
```
output
```
1
```

input
```
4
-1 0 1 0
```
output
```
2
```

input
```
4
1 2 3 -6
```
output
```
3
```

In the first sample, Vasya may transfer 5 from the first bank to the third.

In the second sample, Vasya may first transfer 1 from the third bank to the second, and then 1 from the second to the first.

In the third sample, the following sequence provides the optimal answer:

1. transfer 1 from the first bank to the second bank;
2. transfer 3 from the second bank to the third;
3. transfer 6 from the third bank to the fourth.

D. Soldier and Cards

2 seconds, 256 megabytes

Two bored soldiers are playing card war. Their card deck consists of exactly n cards, numbered from 1 to n, **all values are different**. They divide cards between them in some manner, it's possible that they have different number of cards. Then they play a "war"-like card game.

The rules are following. On each turn a *fight* happens. Each of them picks card from the top of his stack and puts on the table. The one whose card value is bigger wins this *fight* and takes both cards from the table to the bottom of his stack. More precisely, he first takes his opponent's card and puts to the bottom of his stack, and then he puts his card to the bottom of his stack. If after some turn one of the player's stack becomes empty, he loses and the other one wins.

You have to calculate how many *fights* will happen and who will win the game, or state that game won't end.

Input

First line contains a single integer n $(2 \le n \le 10)$, the number of cards.

Second line contains integer k_1 $(1 \le k_1 \le n - 1)$, the number of the first soldier's cards. Then follow k_1 integers that are the values on the first soldier's cards, from top to bottom of his stack.

Third line contains integer k_2 $(k_1 + k_2 = n)$, the number of the second soldier's cards. Then follow k_2 integers that are the values on the second soldier's cards, from top to bottom of his stack.

All card values are different.

Output

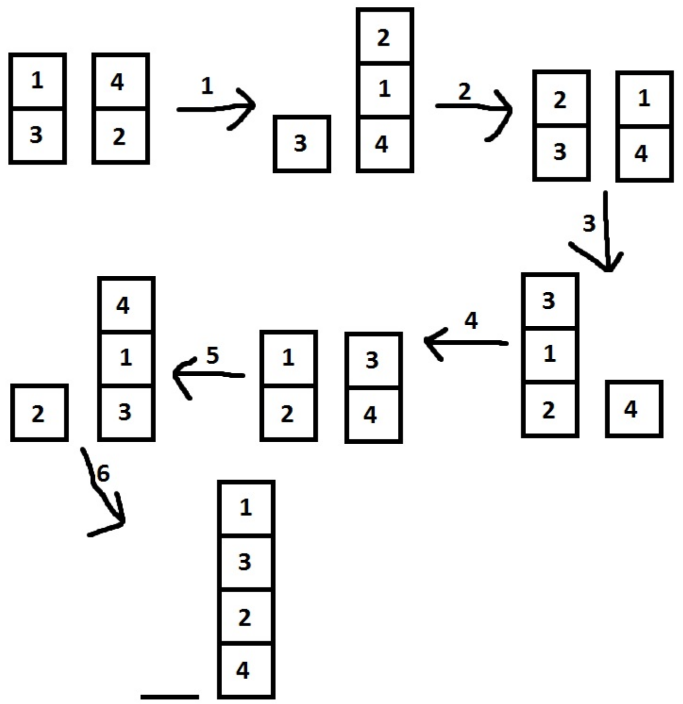
If somebody wins in this game, print 2 integers where the first one stands for the number of *fights* before end of game and the second one is 1 or 2 showing which player has won.

If the game won't end and will continue forever output -1.

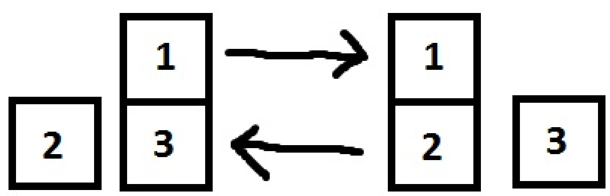
input
```
4
2 1 3
2 4 2
```
output
```
6 2
```

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

First sample:



Second sample:



E. Economy Game

1 second, 256 megabytes

Kolya is developing an economy simulator game. His most favourite part of the development process is in-game testing. Once he was entertained by the testing so much, that he found out his game-coin score become equal to 0.

Kolya remembers that at the beginning of the game his game-coin score was equal to n and that he have bought only some houses (for $1\,234\,567$ game-coins each), cars (for $123\,456$ game-coins each) and computers (for $1\,234$ game-coins each).

Kolya is now interested, whether he could have spent all of his initial n game-coins buying only houses, cars and computers or there is a bug in the game. Formally, is there a triple of non-negative integers a, b and c such that $a \times 1\,234\,567 + b \times 123\,456 + c \times 1\,234 = n$?

Please help Kolya answer this question.

Input

The first line of the input contains a single integer n $(1 \le n \le 10^9)$ — Kolya's initial game-coin score.

Output

Print "YES" (without quotes) if it's possible that Kolya spent all of his initial n coins buying only houses, cars and computers. Otherwise print "NO" (without quotes).

input
```
1359257
```
output
```
YES
```

input
```
17851817
```
output
```
NO
```

In the first sample, one of the possible solutions is to buy one house, one car and one computer, spending $1\,234\,567 + 123\,456 + 1234 = 1\,359\,257$ game-coins in total.

F. Lucky Mask

2 seconds, 256 megabytes

Petya loves lucky numbers very much. Everybody knows that lucky numbers are positive integers whose decimal record contains only the lucky digits **4** and **7**. For example, numbers **47**, **744**, **4** are lucky and **5**, **17**, **467** are not.

Petya calls a mask of a positive integer n the number that is obtained after successive writing of all lucky digits of number n from the left to the right. For example, the mask of number 72174994 is number 7744, the mask of 7 is 7, the mask of 9999047 is 47. Obviously, mask of any number is always a lucky number.

Petya has two numbers — an arbitrary integer a and a lucky number b. Help him find the minimum number c $(c > a)$ such that the mask of number c equals b.

Input

The only line contains two integers a and b $(1 \le a, b \le 10^5)$. It is guaranteed that number b is lucky.

Output

In the only line print a single number — the number c that is sought by Petya.

input
```
1 7
```
output
```
7
```

input
```
100 47
```
output
```
147
```

G. Magic Forest

1 second, 256 megabytes

Imp is in a magic forest, where xorangles grow (wut?)

A xorangle of order n is such a non-degenerate triangle, that lengths of its sides are integers not exceeding n, and the xor-sum of the lengths is equal to zero. Imp has to count the number of distinct xorangles of order n to get out of the forest.

Formally, for a given integer n you have to find the number of such triples (a, b, c), that:

- $1 \le a \le b \le c \le n$;
- $a \oplus b \oplus c = 0$, where $x \oplus y$ denotes the bitwise xor of integers x and y.
- (a, b, c) form a non-degenerate (with strictly positive area) triangle.

Input
The only line contains a single integer n $(1 \le n \le 2500)$.

Output
Print the number of xorangles of order n.

input
```
6
```
output
```
1
```

input
```
10
```
output
```
2
```

The only xorangle in the first sample is $(3, 5, 6)$.

H. Perfect Number

2 seconds, 256 megabytes

We consider a positive integer perfect, if and only if the sum of its digits is exactly 10. Given a positive integer k, your task is to find the k-th smallest perfect positive integer.

Input
A single line with a positive integer k $(1 \leq k \leq 10\,000)$.

Output
A single number, denoting the k-th smallest perfect integer.

input
```
1
```
output
```
19
```

input
```
2
```
output
```
28
```

The first perfect integer is 19 and the second one is 28.

I. Permutations

1 second, 256 megabytes

You are given n k-digit integers. You have to rearrange the digits in the integers so that the difference between the largest and the smallest number was minimum. Digits should be rearranged by the same rule in all integers.

Input
The first line contains integers n and k — the number and digit capacity of numbers correspondingly $(1 \le n, k \le 8)$. Next n lines contain k-digit positive integers. Leading zeroes are allowed both in the initial integers and the integers resulting from the rearranging of digits.

Output
Print a single number: the minimally possible difference between the largest and the smallest number after the digits are rearranged in all integers by the same rule.

input
```
6 4
5237
2753
7523
5723
5327
2537
```
output
```
2700
```

input
```
3 3
010
909
012
```
output
```
3
```

input
```
7 5
50808
36603
37198
44911
29994
42543
50156
```
output
```
20522
```

In the first sample, if we rearrange the digits in numbers as (3,1,4,2), then the 2-nd and the 4-th numbers will equal 5237 and 2537 correspondingly (they will be maximum and minimum for such order of digits).

In the second sample, if we swap the second digits and the first ones, we get integers 100, 99 and 102.

J. Minimum Sum

1 second, 256 megabytes

Petya has n positive integers $a_1, a_2, ..., a_n$.

His friend Vasya decided to joke and replaced all digits in Petya's numbers with a letters. He used the lowercase letters of the Latin alphabet from 'a' to 'j' and replaced all digits 0 with one letter, all digits 1 with another letter and so on. For any two different digits Vasya used distinct letters from 'a' to 'j'.

Your task is to restore Petya's numbers. The restored numbers should be **positive integers** without leading zeros. Since there can be multiple ways to do it, determine the **minimum** possible sum of all Petya's numbers after the restoration. It is guaranteed that before Vasya's joke all Petya's numbers did not have leading zeros.

Input
The first line contains a single integer n $(1 \le n \le 1\,000)$ — the number of Petya's numbers.

Each of the following lines contains non-empty string s_i consisting of lowercase Latin letters from 'a' to 'j' — the Petya's numbers after Vasya's joke. The length of each string does not exceed six characters.

Output

Determine the **minimum** sum of all Petya's numbers after the restoration. The restored numbers should be **positive integers** without leading zeros. It is guaranteed that the correct restore (without leading zeros) exists for all given tests.

input
```
3
ab
de
aj
```
output
```
47
```

input
```
5
abcdef
ghij
bdef
accbd
g
```
output
```
136542
```

input
```
3
aa
jj
aa
```
output
```
44
```

In the first example, you need to replace the letter 'a' with the digit 1, the letter 'b' with the digit 0, the letter 'd' with the digit 2, the letter 'e' with the digit 3, and the letter 'j' with the digit 4. So after the restoration numbers will look like $[10, 23, 14]$. The sum of them is equal to 47, which is the minimum possible sum of the numbers after the correct restoration.

In the second example the numbers after the restoration can look like: $[120468, 3579, 2468, 10024, 3]$.

In the second example the numbers after the restoration can look like: $[11, 22, 11]$.

K. Drazil and His Happy Friends

2 seconds, 256 megabytes

Drazil has many friends. Some of them are happy and some of them are unhappy. Drazil wants to make all his friends become happy. So he invented the following plan.

There are n boys and m girls among his friends. Let's number them from 0 to $n - 1$ and 0 to $m - 1$ separately. In i-th day, Drazil invites $(i \bmod n)$-th boy and $(i \bmod m)$-th girl to have dinner together (as Drazil is programmer, i starts from 0). If one of those two people is happy, the other one will also become happy. Otherwise, those two people remain in their states. Once a person becomes happy (or if he/she was happy originally), he stays happy forever.

Drazil wants to know whether he can use this plan to make all his friends become happy at some moment.

Input

The first line contains two integer n and m $(1 \le n, m \le 100)$.

The second line contains integer b $(0 \le b \le n)$, denoting the number of happy boys among friends of Drazil, and then follow b distinct integers $x_1, x_2, ..., x_b$ $(0 \le x_i < n)$, denoting the list of indices of happy boys.

The third line conatins integer g $(0 \le g \le m)$, denoting the number of happy girls among friends of Drazil, and then follow g distinct integers $y_1, y_2, ... , y_g$ $(0 \le y_j < m)$, denoting the list of indices of happy girls.

It is guaranteed that there is at least one person that is unhappy among his friends.

Output

If Drazil can make all his friends become happy by this plan, print "Yes". Otherwise, print "No".

input
```
2 3
0
1 0
```
output
```
Yes
```

input
```
2 4
1 0
1 2
```
output
```
No
```

input
```
2 3
1 0
1 1
```
output
```
Yes
```

By $i \bmod k$ we define the remainder of integer division of i by k.

In first sample case:

- On the 0-th day, Drazil invites 0-th boy and 0-th girl. Because 0-th girl is happy at the beginning, 0-th boy become happy at this day.
- On the 1-st day, Drazil invites 1-st boy and 1-st girl. They are both unhappy, so nothing changes at this day.
- On the 2-nd day, Drazil invites 0-th boy and 2-nd girl. Because 0-th boy is already happy he makes 2-nd girl become happy at this day.
- On the 3-rd day, Drazil invites 1-st boy and 0-th girl. 0-th girl is happy, so she makes 1-st boy happy.
- On the 4-th day, Drazil invites 0-th boy and 1-st girl. 0-th boy is happy, so he makes the 1-st girl happy. So, all friends become happy at this moment.

L. Optimal Point on a Line

1 second, 256 megabytes

You are given n points on a line with their coordinates x_i. Find the point x so the sum of distances to the given points is minimal.

Input

The first line contains integer n $(1 \le n \le 3 \cdot 10^5)$ — the number of points on the line.

The second line contains n integers x_i $(-10^9 \le x_i \le 10^9)$ — the coordinates of the given n points.

Output

Print the only integer x — the position of the optimal point on the line. If there are several optimal points print the position of the leftmost one. It is guaranteed that the answer is always the integer.

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

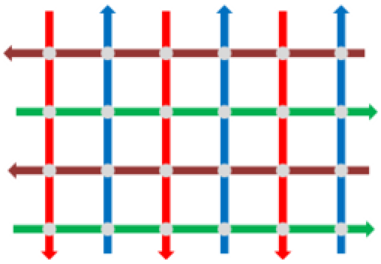
output

```
2
```

M. Strongly Connected City

2 seconds, 256 megabytes

Imagine a city with n horizontal streets crossing m vertical streets, forming an $(n - 1) \times (m - 1)$ grid. In order to increase the traffic flow, mayor of the city has decided to make each street one way. This means in each horizontal street, the traffic moves only from west to east or only from east to west. Also, traffic moves only from north to south or only from south to north in each vertical street. It is possible to enter a horizontal street from a vertical street, or vice versa, at their intersection.



The mayor has received some street direction patterns. Your task is to check whether it is possible to reach any junction from any other junction in the proposed street direction pattern.

Input

The first line of input contains two integers n and m, $(2 \le n, m \le 20)$, denoting the number of horizontal streets and the number of vertical streets.

The second line contains a string of length n, made of characters '<' and '>', denoting direction of each horizontal street. If the i-th character is equal to '<', the street is directed from east to west otherwise, the street is directed from west to east. Streets are listed in order from north to south.

The third line contains a string of length m, made of characters '^' and 'v', denoting direction of each vertical street. If the i-th character is equal to '^', the street is directed from south to north, otherwise the street is directed from north to south. Streets are listed in order from west to east.

Output

If the given pattern meets the mayor's criteria, print a single line containing "YES", otherwise print a single line containing "`NO`".

input

```
3 3
><>
v^v
```

output

```
NO
```

input

```
4 6
<><>
v^v^v^
```

output

```
YES
```

The figure above shows street directions in the second sample test case.

N. Multiplication Table

1 second, 256 megabytes

Bizon the Champion isn't just charming, he also is very smart.

While some of us were learning the multiplication table, Bizon the Champion had fun in his own manner. Bizon the Champion painted an $n \times m$ multiplication table, where the element on the intersection of the i-th row and j-th column equals $i \cdot j$ (the rows and columns of the table are numbered starting from 1). Then he was asked: what number in the table is the k-th largest number? Bizon the Champion always answered correctly and immediately. Can you repeat his success?

Consider the given multiplication table. If you write out all $n \cdot m$ numbers from the table in the non-decreasing order, then the k-th number you write out is called the k-th largest number.

Input

The single line contains integers n, m and k $(1 \le n, m \le 5 \cdot 10^5; 1 \le k \le n \cdot m)$.

Output

Print the k-th largest number in a $n \times m$ multiplication table.

input

```
2 2 2
```

output

```
2
```

input

```
2 3 4
```

output

```
3
```

input

```
1 10 5
```

output

```
5
```

A 2×3 multiplication table looks like this:

```
1 2 3
2 4 6
```

O. Antipalindrome

1 second, 256 megabytes

A string is a palindrome if it reads the same from the left to the right and from the right to the left. For example, the strings "`kek`", "`abacaba`", "`r`" and "`papicipap`" are palindromes, while the strings "`abb`" and "`iq`" are not.

A substring $s[l \ldots r]$ $(1 \leq l \leq r \leq |s|)$ of a string $s = s_1 s_2 \ldots s_{|s|}$ is the string $s_l s_{l+1} \ldots s_r$.

Anna does not like palindromes, so she makes her friends call her Ann. She also changes all the words she reads in a similar way. Namely, each word s is changed into its longest substring that is not a palindrome. If all the substrings of s are palindromes, she skips the word at all.

Some time ago Ann read the word s. What is the word she changed it into?

Input

The first line contains a non-empty string s with length at most 50 characters, containing lowercase English letters only.

Output

If there is such a substring in s that is not a palindrome, print the maximum length of such a substring. Otherwise print 0.

Note that there can be multiple longest substrings that are not palindromes, but their length is unique.

input
```
mew
```
output
```
3
```

input
```
wuffuw
```
output
```
5
```

input
```
qqqqqqqq
```
output
```
0
```

"`mew`" is not a palindrome, so the longest substring of it that is not a palindrome, is the string "`mew`" itself. Thus, the answer for the first example is 3.

The string "`uffuw`" is one of the longest non-palindrome substrings (of length 5) of the string "`wuffuw`", so the answer for the second example is 5.

All substrings of the string "`qqqqqqqq`" consist of equal characters so they are palindromes. This way, there are no non-palindrome substrings. Thus, the answer for the third example is 0.

P. Magic Powder - 1

1 second, 256 megabytes

This problem is given in two versions that differ only by constraints. If you can solve this problem in large constraints, then you can just write a single solution to the both versions. If you find the problem too difficult in large constraints, you can write solution to the simplified version only.

Waking up in the morning, Apollinaria decided to bake cookies. To bake one cookie, she needs n ingredients, and for each ingredient she knows the value a_i — how many grams of this ingredient one needs to bake a cookie. To prepare one cookie Apollinaria needs to use all n ingredients.

Apollinaria has b_i gram of the i-th ingredient. Also she has k grams of a magic powder. Each gram of magic powder can be turned to exactly 1 gram of any of the n ingredients and can be used for baking cookies.

Your task is to determine the maximum number of cookies, which Apollinaria is able to bake using the ingredients that she has and the magic powder.

Input
The first line of the input contains two positive integers n and k $(1 \le n, k \le 1000)$ — the number of ingredients and the number of grams of the magic powder.

The second line contains the sequence $a_1, a_2, ..., a_n$ $(1 \le a_i \le 1000)$, where the i-th number is equal to the number of grams of the i-th ingredient, needed to bake one cookie.

The third line contains the sequence $b_1, b_2, ..., b_n$ $(1 \le b_i \le 1000)$, where the i-th number is equal to the number of grams of the i-th ingredient, which Apollinaria has.

Output
Print the maximum number of cookies, which Apollinaria will be able to bake using the ingredients that she has and the magic powder.

input
```
3 1
2 1 4
11 3 16
```
output
```
4
```

input
```
4 3
4 3 5 6
11 12 14 20
```
output
```
3
```

In the first sample it is profitably for Apollinaria to make the existing 1 gram of her magic powder to ingredient with the index 2, then Apollinaria will be able to bake 4 cookies.

In the second sample Apollinaria should turn 1 gram of magic powder to ingredient with the index 1 and 1 gram of magic powder to ingredient with the index 3. Then Apollinaria will be able to bake 3 cookies. The remaining 1 gram of the magic powder can be left, because it can't be used to increase the answer.

Q. DZY Loves Chessboard

1 second, 256 megabytes

DZY loves chessboard, and he enjoys playing with it.

He has a chessboard of n rows and m columns. Some cells of the chessboard are bad, others are good. For every good cell, DZY wants to put a chessman on it. Each chessman is either white or black. After putting all chessmen, DZY wants that no two chessmen with the same color are on two adjacent cells. Two cells are adjacent if and only if they share a common edge.

You task is to find any suitable placement of chessmen on the given chessboard.

Input
The first line contains two space-separated integers n and m $(1 \le n, m \le 100)$.

Each of the next n lines contains a string of m characters: the j-th character of the i-th string is either "`.`" or "`-`". A "`.`" means that the corresponding cell (in the i-th row and the j-th column) is good, while a "`-`" means it is bad.

Output
Output must contain n lines, each line must contain a string of m characters. The j-th character of the i-th string should be either "`W`", "B" or "`-`". Character "`W`" means the chessman on the cell is white, "B" means it is black, "`-`" means the cell is a bad cell.

If multiple answers exist, print any of them. It is guaranteed that at least one answer exists.

input
```
1 1
.
```
output
```
B
```

input
```
2 2
..
..
```
output
```
BW
WB
```

input
```
3 3
.-.
---
--.
```

output

```
B-B
---
--B
```

In the first sample, DZY puts a single black chessman. Of course putting a white one is also OK.

In the second sample, all 4 cells are good. No two same chessmen share an edge in the sample output.

In the third sample, no good cells are adjacent. So you can just put 3 chessmen, no matter what their colors are.

R. Nineteen

1 second, 256 megabytes

Alice likes word "`nineteen`" very much. She has a string s and wants the string to contain as many such words as possible. For that reason she can rearrange the letters of the string.

For example, if she has string "`xiineteenppnnnewtnee`", she can get string "`x`**`nineteen`**`pp`**`nineteen`**`w`", containing (the occurrences marked) two such words. More formally, word "`nineteen`" occurs in the string the number of times you can read it starting from some letter of the string. Of course, you shouldn't skip letters.

Help her to find the maximum number of "`nineteen`"s that she can get in her string.

Input

The first line contains a non-empty string s, consisting only of lowercase English letters. The length of string s doesn't exceed 100.

Output

Print a single integer — the maximum number of "`nineteen`"s that she can get in her string.

input

```
nniinneetteeeenn
```

output

```
2
```

input

```
nneteenabcnneteenabcnneteenabcnneteenabcnneteenabcii
```

output

```
2
```

input

```
nineteenineteen
```

output

```
2
```

S. Chess For Three

1 second, 256 megabytes

Alex, Bob and Carl will soon participate in a team chess tournament. Since they are all in the same team, they have decided to practise really hard before the tournament. But it's a bit difficult for them because chess is a game for two players, not three.

So they play with each other according to following rules:

- Alex and Bob play the first game, and Carl is spectating;
- When the game ends, the one who lost the game becomes the spectator in the next game, and the one who was spectating plays against the winner.

Alex, Bob and Carl play in such a way that there are no draws.

Today they have played n games, and for each of these games they remember who was the winner. They decided to make up a log of games describing who won each game. But now they doubt if the information in the log is correct, and they want to know if the situation described in the log they made up was possible (that is, no game is won by someone who is spectating if Alex, Bob and Carl play according to the rules). Help them to check it!

Input

The first line contains one integer n $(1 \le n \le 100)$ — the number of games Alex, Bob and Carl played.

Then n lines follow, describing the game log. i-th line contains one integer a_i $(1 \le a_i \le 3)$ which is equal to 1 if Alex won i-th game, to 2 if Bob won i-th game and 3 if Carl won i-th game.

Output

Print YES if the situation described in the log was possible. Otherwise print `NO`.

input

```
3
1
1
2
```

output

```
YES
```

input

```
2
1
2
```

output

```
NO
```

In the first example the possible situation is:

1. Alex wins, Carl starts playing instead of Bob;
2. Alex wins, Bob replaces Carl;
3. Bob wins.

The situation in the second example is impossible because Bob loses the first game, so he cannot win the second one.

T. Greed

2 seconds, 256 megabytes

Jafar has n cans of cola. Each can is described by two integers: remaining volume of cola a_i and can's capacity b_i ($a_i \le b_i$).

Jafar has decided to pour all remaining cola into just 2 cans, determine if he can do this or not!

Input

The first line of the input contains one integer n $(2 \le n \le 100\,000)$ — number of cola cans.

The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^9)$ — volume of remaining cola in cans.

The third line contains n space-separated integers that $b_1, b_2, ..., b_n$ ($a_i \le b_i \le 10^9$) — capacities of the cans.

Output

Print "YES" (without quotes) if it is possible to pour all remaining cola in 2 cans. Otherwise print "`NO`" (without quotes).

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

input

```
2
3 5
3 6
```

output
```
YES
```

input
```
3
6 8 9
6 10 12
```
output
```
NO
```

input
```
5
0 0 5 0 0
1 1 8 10 5
```
output
```
YES
```

input
```
4
4 1 0 3
5 2 2 3
```
output
```
YES
```

In the first sample, there are already 2 cans, so the answer is "YES".

U. Local Extrema

1 second, 256 megabytes

You are given an array a. Some element of this array a_i is a *local minimum* iff it is strictly less than both of its neighbours (that is, $a_i < a_{i-1}$ and $a_i < a_{i+1}$). Also the element can be called *local maximum* iff it is strictly greater than its neighbours (that is, $a_i > a_{i-1}$ and $a_i > a_{i+1}$). Since a_1 and a_n have only one neighbour each, they are neither local minima nor local maxima.

An element is called a *local extremum* iff it is either local maximum or local minimum. Your task is to calculate the number of local extrema in the given array.

Input

The first line contains one integer n $(1 \le n \le 1000)$ — the number of elements in array a.

The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 1000)$ — the elements of array a.

Output

Print the number of local extrema in the given array.

input
```
3
1 2 3
```
output
```
0
```

input
```
4
1 5 2 5
```
output
```
2
```

V. Scarborough Fair

2 seconds, 256 megabytes

Are you going to Scarborough Fair?

Parsley, sage, rosemary and thyme.

Remember me to one who lives there.

He once was the true love of mine.

Willem is taking the girl to the highest building in island No.28, however, neither of them knows how to get there.

Willem asks his friend, Grick for directions, Grick helped them, and gave them a task.

Although the girl wants to help, Willem insists on doing it by himself.

Grick gave Willem a string of length n.

Willem needs to do m operations, each operation has four parameters l, r, c_1, c_2, which means that all symbols c_1 in range $[l, r]$ (from l-th to r-th, including l and r) are changed into c_2. String is 1-indexed.

Grick wants to know the final string after all the m operations.

Input

The first line contains two integers n and m $(1 \le n, m \le 100)$.

The second line contains a string s of length n, consisting of lowercase English letters.

Each of the next m lines contains four parameters l, r, c_1, c_2 $(1 \le l \le r \le n, c_1, c_2$ are lowercase English letters), separated by space.

Output

Output string s after performing m operations described above.

input
```
3 1
ioi
1 1 i n
```
output
```
noi
```

input
```
5 3
wxhak
3 3 h x
1 5 x a
1 3 w g
```
output
```
gaaak
```

For the second example:

After the first operation, the string is `wxxak`.

After the second operation, the string is `waaak`.

After the third operation, the string is `gaaak`.

W. Div. 64

1 second, 256 megabytes

Top-model Izabella participates in the competition. She wants to impress judges and show her mathematical skills.

Her problem is following: for given string, consisting of only 0 and 1, tell if it's possible to remove some digits in such a way, that remaining number is a representation of some positive integer, divisible by 64, in the binary numerical system.

Input

In the only line given a non-empty binary string s with length up to 100.

Output

Print «`yes`» (without quotes) if it's possible to remove digits required way and «`no`» otherwise.

input
100010001
output
yes

input
100
output
no

In the first test case, you can get string $1\,000\,000$ after removing two ones which is a representation of number 64 in the binary numerical system.

You can read more about binary numeral system representation here: https://en.wikipedia.org/wiki/Binary_system

X. Book Reading

2 seconds, 256 megabytes

Recently Luba bought a very interesting book. She knows that it will take t seconds to read the book. Luba wants to finish reading as fast as she can.

But she has some work to do in each of n next days. The number of seconds that Luba has to spend working during i-th day is a_i. If some free time remains, she can spend it on reading.

Help Luba to determine the minimum number of day when she finishes reading.

It is guaranteed that the answer doesn't exceed n.

Remember that there are 86400 seconds in a day.

Input

The first line contains two integers n and t ($1 \le n \le 100$, $1 \le t \le 10^6$) — the number of days and the time required to read the book.

The second line contains n integers a_i ($0 \le a_i \le 86400$) — the time Luba has to spend on her work during i-th day.

Output

Print the minimum day Luba can finish reading the book.

It is guaranteed that answer doesn't exceed n.

input
2 2 86400 86398
output
2

input
2 86400 0 86400
output
1

Y. Chtholly's request

2 seconds, 256 megabytes

— Thanks a lot for today.
— I experienced so many great things.

— You gave me memories like dreams... But I have to leave now...

— One last request, can you...

— Help me solve a Codeforces problem?

— ......

— What?

Chtholly has been thinking about a problem for days:

If a number is *palindrome* and length of its decimal representation without leading zeros is even, we call it a zcy number. A number is *palindrome* means when written in decimal representation, it contains no leading zeros and reads the same forwards and backwards. For example `12321` and `1221` are palindromes and `123` and `12451` are not. Moreover, `1221` is zcy number and `12321` is not.

Given integers k and p, calculate the sum of the k smallest zcy numbers and output this sum modulo p.

Unfortunately, Willem isn't good at solving this kind of problems, so he asks you for help!

Input

The first line contains two integers k and p ($1 \le k \le 10^5$, $1 \le p \le 10^9$).

Output

Output single integer — answer to the problem.

input
2 100
output
33

input
5 30
output
15

In the first example, the smallest zcy number is 11, and the second smallest zcy number is 22.

In the second example, $(11 + 22 + 33 + 44 + 55) \bmod 30 = 15$.

Z. Godsend

2 seconds, 256 megabytes

Leha somehow found an array consisting of n integers. Looking at it, he came up with a task. Two players play the game on the array. Players move one by one. The first player can choose for his move a subsegment of non-zero length with an odd sum of numbers and remove it from the array, after that the remaining parts are glued together into one array and the game continues. The second player can choose a subsegment of non-zero length with an even sum and remove it. Loses the one who can not make a move. Who will win if both play optimally?

Input

First line of input data contains single integer n ($1 \le n \le 10^6$) — length of the array.

Next line contains n integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le 10^9$).

Output

Output answer in single line. "`First`", if first player wins, and "`Second`" otherwise (without quotes).

input
4 1 3 2 3
output
First

input

```
2
2 2
```

output

```
Second
```

In first sample first player remove whole array in one move and win.

In second sample first player can't make a move and lose.