A. Team

One day three best friends Petya, Vasya and Tonya decided to form a team and take part in programming contests. Participants are usually offered several problems during programming contests. Long before the start the friends decided that they will implement a problem if at least two of them are sure about the solution. Otherwise, the friends won't write the problem's solution.

This contest offers n problems to the participants. For each problem we know, which friend is sure about the solution. Help the friends find the number of problems for which they will write a solution.

Input

The first input line contains a single integer n ($1 \le n \le 1000$) — the number of problems in the contest. Then n lines contain three integers each, each integer is either 0 or 1. If the first number in the line equals 1, then Petya is sure about the problem's solution, otherwise he isn't sure. The second number shows Vasya's view on the solution, the third number shows Tonya's view. The numbers on the lines are separated by spaces.

Output

Print a single integer — the number of problems the friends will implement on the contest.

Examples

Sample Input 1

```
3
1 1 0
1 1 1
1 0 0
```

Sample Output 1

```
2
```

Input

```
2
1 0 0
0 1 1
```

Output

```
1
```

Note

In the first sample Petya and Vasya are sure that they know how to solve the first problem and all three of them know how to solve the second problem. That means that they will write solutions for these problems. Only Petya is sure about the solution for the third problem, but that isn't enough, so the friends won't take it.

In the second sample the friends will only implement the second problem, as Vasya and Tonya are sure about the solution.

B. Watermelon

On a hot summer day, Pete and his friend Billy decided to buy a watermelon. After choosing the biggest and ripest one, they found that it weighs **w** kilos. Tired and thirsty, they hurried home and decided to split it before eating.

Pete and Billy like even numbers, so they want to divide the watermelon into **two parts of even weight** . The two parts do not have to be equal, but **both must have a positive weight** . Help them determine whether such a division is possible.

Input The first (and the only) input line contains integer number w ($1 \le w \le 100$) — the weight of the watermelon bought by the boys.

Output Print YES, if the boys can divide the watermelon into two parts, each of them weighing even number of kilos; and NO in the opposite case.

Input

8

Output

YES

Note: For example, the boys can divide the watermelon into two parts of 2 and 6 kilos respectively (another variant — two parts of 4 and 4 kilos).

C. Borze

Ternary numeric notation is quite popular in Berland. To telegraph the ternary number the Borze alphabet is used. Digit 0 is transmitted as «.», 1 as «-.» and 2 as «--». You are to decode the Borze code, i.e. to find out the ternary number given its representation in Borze alphabet.

Input

The first line contains a number in Borze code. The length of the string is between 1 and 200 characters. It's guaranteed that the given string is a valid Borze code of some ternary number (this number can have leading zeroes).

Output

Output the decoded ternary number. It can have leading zeroes.

Examples

Input	Output
.-.--	012
--.	20
-..-.--	1012

D. Translation

The translation from the Berland language into the Birland language is not an easy task. Those languages are very similar: a Berlandish word differs from a Birlandish word with the same meaning a little: it is spelled (and pronounced) reversely. For example, a Berlandish word code corresponds to a Birlandish word edoc. However, making a mistake during the "translation" is easy. Vasya translated the word *s* from Berlandish into Birlandish as *t*. Help him: find out if he translated the word correctly.

Input

The first line contains word *s*, the second line contains word *t*. The words consist of lowercase Latin letters. The input data do not contain unnecessary spaces. The words are not empty and their lengths do not exceed 100 symbols.

Output

If the word *t* is a word *s*, written reversely, print YES, otherwise print NO.

Examples

Input	Output
code edoc	YES

Input	Output
abb aba	NO

Input	Output
code code	NO

E. Lucky Ticket

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 loves tickets very much. As we know, each ticket has a number that is a positive integer. Its length equals *n* (*n* is always even). Petya calls a ticket lucky if the ticket's number is a lucky number and the sum of digits in the first half (the sum of the first *n* / 2 digits) equals the sum of digits in the second half (the sum of the last *n* / 2 digits). Check if the given ticket is lucky.

Input

The first line contains an even integer n ($2 \le n \le 50$) — the length of the ticket number that needs to be checked. The second line contains an integer whose length equals exactly *n* — the ticket number. The number may contain leading zeros.

Output

On the first line print "YES" if the given ticket number is lucky. Otherwise, print "NO" (without the quotes).

Examples

Input

```
2
47
```

Output

```
NO
```

Input

```
4
4738
```

Output

```
NO
```

Input

```
4
4774
```

Output

```
YES
```

Note

In the first sample the sum of digits in the first half does not equal the sum of digits in the second half (4 ≠ 7).

In the second sample the ticket number is not the lucky number.

F. Winning ICPC

There are N teams (numbered from 1 to N) and problems (numbered from 1 to) in this year's ICPC. The j-th problem has T_j testcases. Surprisingly, every team submitted exactly one solution to every problem. The -th team managed to solve $S_{i,j}$ testcases on the j-th problem.

A team solved a problem only if the team managed to solve ALL testcases on that problem. The winning team is the team with the most number of problems solved. If there are more than one team with the most number of problems solved, then the winning team is the team with the smallest index among those teams.

Determine the index of the winning team.

Input

The first line contains two integers: N (1 ≤ N, ≤ 100) in a line denoting the number of teams and

the number of problems. The second line contains integers: T_1 T_2 T_M (0 ≤ T_i ≤ 100) in a line denoting the number of testcases. The next N following lines, each contains integers; the j-th integer on the -th line is $S_{i,j}$ (0 ≤ $S_{i,j}$ ≤ T_j) denoting the number of solved testcases by the -th team for the j-th problem.

Output

The output contains the index of the winning team, in a line.

Sample Input	Output for Sample Input
3 2 10 20 0 19 10 0 9 19	2
3 2 10 20 0 20 10 0 9 19	1
1 1 1 0	1

Explanation for the 1st sample case

On the first sample, the first and the third team did not solve any problem, and the second teamsolved the first problem. Therefore, the second team is the winner.

Explanation for the 2nd sample case

On the second sample, the first team solved the second problem, the second team solved the first problem, and the third team did not solve any problem. Since the first team has a smaller index than the second team, the first team is the winner.

Explanation for the 3rd sample case

On the third sample, there is only one team thus the winner is obvious.

G. Lie Detector

Andi is a young and prominent detective in the police force. His ability to track down criminals, uncover the truth, and solve cases never ceases to amaze all of his colleagues. One day, he is faced with a suspicious eyewitness testimony when working on a certain case. In usual cases, Andi simply ignores such unreliable testimony; however, in this case, the eyewitness testimony is too important to be ignored. To resolve this situation, Andi has to rely on technology, i.e. using a lie detector.

Andi proceeds to use a lie detector to detect whether the eyewitness testimony is true. However, Andi notices that the lie detector he used might have been tampered, thus, he employs a second lie detector to detect whether the first lie detector's result is correct. This situation happens repeatedly such that Andi ends up employing N lie detectors in total. The i^{th} lie detector reports the truth of the $(i-1)^{th}$ lie detector for $i = 2..N$, and the 1^{st} lie detector reports the truth of the eyewitness testimony.

In the end, Andi knows that the last (N^{th}) lie detector has not been tampered and always report the truth correctly. Now, he needs to determine whether the eyewitness testimony is true given the result of all lie detectors.

For example, let $N = 4$ and the lie detectors result are (LIE, LIE, TRUTH, TRUTH

).

- The 4^{th} lie detector reports that the 3^{rd} lie detector is TRUTH. As the 4^{th} lie detector always report the truth correctly, then the 3^{rd} lie detector's result is correct as it is.
- The 3^{rd} lie detector reports that the 2^{nd} lie detector is TRUTH. As the 3^{rd} lie detector's result is correct as it is, then the 2^{nd} lie detector's result is also correct as it is.
- The 2^{nd} lie detector reports that the 1^{st} lie detector is LIE. As the 2^{nd} lie detector's result is correct as it is, then the 1^{st} lie detector's result is wrong.
- The 1^{st} lie detector reports that the eyewitness testimony is LIE. As the 1^{st} lie detector's result is wrong, then the eyewitness testimony is correct; in other words, what the eyewitness says is true.

Therefore, the eyewitness testimony in this example is true.

Input

Input begins with a line containing an integer N ($2 \le N \le 100000$). The next N lines, each contains a string S_i (either TRUTH or LIE) representing the output of the i^{th} lie detector for $i = 1..N$ respectively.

Output

Output contains a string TRUTH or LIE in a line whether the eyewitness testimony is true or false.

Sample Input #1

```
4
LIE
LIE
TRUTH
TRUTH
```

Sample Output #1

```
TRUTH
```

Explanation for the sample input/output #1

This sample is illustrated in the problem description above.

Sample Input #2

```
3
LIE
LIE
LIE
```

Sample Output #2

```
LIE
```

H. Uniform Maker

The International Costumes and Props Company (ICPC) received an order from a client to produce N pennants each containing the same word. However, due to some miscommunication between the account manager and the client, not all the produced pennants have the same word although all of them have a word of the same length. Reproducing those pennants is very costly as the ICPC only uses a certain type of rare fabric in their production.

Fortunately, the client didn't specify the word that they want to be in the pennants. In fact, the client will be satisfied if and only if all the pennants have the same word.

The ICPC has a special technique to change one character in a word into some other character. It is ex- pensive, albeit not as expensive as reproducing a new pennant. Therefore, the ICPC has to minimize the number of times they have to use such a technique. Your task in this problem is to help the ICPC to determine the minimum total number of characters that need to be changed so that the client will be satisfied.

For example, let there be $N = 6$ pennants with the following words: calf, palm, book, icpc, ball, and room. The total number of characters than need to be changed can be minimized if all the words are changed into balm.

- calf → 2 characters: b**m
- palm → 1 characters: b***
- book → 3 characters: *alm
- icpc → 4 characters: balm
- ball → 1 characters: ***m
- room → 3 characters: bal*

The symbol * represents an unchanged character. There are a total of 14 characters that need to be changed in this example.

Input

Input begins with a line containing two integers N M ($2 \le N \le 100$; $1 \le M \le 100$) representing the number of pennants and the length of each word in the pennant, respectively. The next N line each contains a string S_i ($|S_i| = M$) representing the word on the i^{th} pennant. Each string only contains lowercase alphabetical characters.

Output

Output contains an integer in a line representing the minimum total number of characters that need to be changed so that the

Sample Input #1

```
6 4
calf
palm
book
icpc
ball
room
```

Sample Output #1

```
14
```

Explanation for the sample input/output #1

This is the example from the problem description.

Sample Input #2

```
3 11
goodluckfor
icpcjakarta
contestants
```

Sample Output #2

```
19
```

Sample Input #3

```
5 14
helpiamtrapped
inanincfactory
forthreemonths
withoutfoodand
drinkandshower
```

Sample Output #3

```
49
```

I. Exchange Bottleneck

The country of Bazbesonin currently has N cities (numbered from 1 to N) connected by bidirectional roads. When a pair of cities, u and v, would like to exchange a message, the **latency** is defined as the minimum number of roads required to go from u to v.

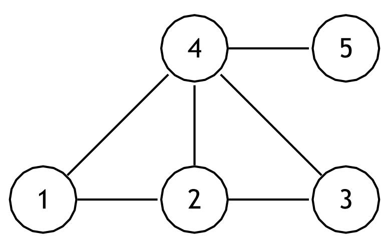
These cities have a long history in the past. Initially, city 1 is constructed in the middle of Bazbesonin. Thereafter, the rest of the cities are constructed one after another from city 2 to city N. When constructing city x, one or more bidirectional roads were also constructed depending on the economic condition of Bazbesonin at that time.

- If city x was constructed when the economy was good, the roads connecting city x and all cities constructed previously were constructed. In other words, the roads connecting city x and city y were constructed, for all $1 \leq y < x$.
- If city x was constructed when the economy was bad, only the road connecting city x and city $x - 1$ was constructed.

The economic condition of Bazbesonin is represented by the binary array $E_{1 \ldots N-1}$. If the economy was good when city x was constructed, then the value of E_{x-1} is 1. Otherwise, the value of E_{x-1} is 0.

Back to the present day, each of the N cities would like to exchanges a message with every other city. The bottleneck of the exchange is the maximum latency among all pairs of cities. We would like to compute the bottleneck of the message exchange.

For example, let $N = 5$ and $B_{1 \ldots 4} = [1, 0, 1, 0]$. The cities and roads in Bazbesonin can be illustrated by the following figure:



- The latency of city 1 and city 2 is 1.
- The latency of city 1 and city 3 is 2.
- The latency of city 1 and city 4 is 1.
- The latency of city 1 and city 5 is 2.
- The latency of city 2 and city 3 is 1

- The latency of city 2 and city 4 is 1.
- The latency of city 2 and city 5 is 2.
- The latency of city 3 and city 4 is 1.
- The latency of city 3 and city 5 is 2.
- The latency of city 4 and city 5 is 1.

Therefore, the bottleneck in this example is 2.

Input

Input begins with a line containing an integer: N ($2 \le N \le 100\,000$) representing the number of cities in Bazbesonin. The next line contains $N - 1$ integers: E_i ($E_i \in \{0, 1\}$) representing the economic condition of Bazbesonin.

Output

Output in a line an integer representing the bottleneck of the message exchange.

Sample Input #1

```
5
1 0 1 0
```

Sample Output #1

```
2
```

Explanation for the sample input/output #1

This is the example from the problem description.

Sample Input #2

```
7
1 1 1 1 1 1
```

Sample Output #2

```
1
```

Explanation for the sample input/output #1

Each pair of cities is connected by a road, and thus their latencies are 1.