

C. Removing Columns

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given an $n \times m$ rectangular table consisting of lower case English letters. In one operation you can completely remove one column from the table. The remaining parts are combined forming a new table. For example, after removing the second column from the table

```
abcd
edfg
hijk
```

we obtain the table:

```
acd
efg
hjk
```

A table is called *good* if its rows are ordered from top to bottom lexicographically, i.e. each row is lexicographically no larger than the following one. Determine the minimum number of operations of removing a column needed to make a given table good.

Input

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

Next n lines contain m small English letters each — the characters of the table.

Output

Print a single number — the minimum number of columns that you need to remove in order to make the table good.

Examples

input
1 10 codeforces
output
0
input
4 4 case care test code
output
2
input
5 4 code forc esco

Codeforces Round #283 (Div. 2)

Finished

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.



Start virtual contest

→ Problem tags

brute force constructive algorithms
implementation

No tag edit access

→ Contest materials

- Announcement 
- Tutorial 

defo rces
output
4

Note

In the first sample the table is already good.

In the second sample you may remove the first and third column.

In the third sample you have to remove all the columns (note that the table where all rows are empty is considered good by definition).

Let strings s and t have equal length. Then, s is *lexicographically larger* than t if they are not equal and the character following the largest common prefix of s and t (the prefix may be empty) in s is alphabetically larger than the corresponding character of t .