A set of N words is given with the length of each word being exactly **2K** characters.

A **directed graph** with each vertex containing a single letter is called a "**kokos**" if, for **each** word in the set, there **exists a directed path** in the graph such that the labels on the vertices along that path **form the word**. Additionally, for all vertices on that path the following conditions have to be satisfied:

· the in-degree of the first vertex is 0

· the in-degrees of the next K-1 vertices is 1

· the out-degrees of the next K-1 vertices is 1

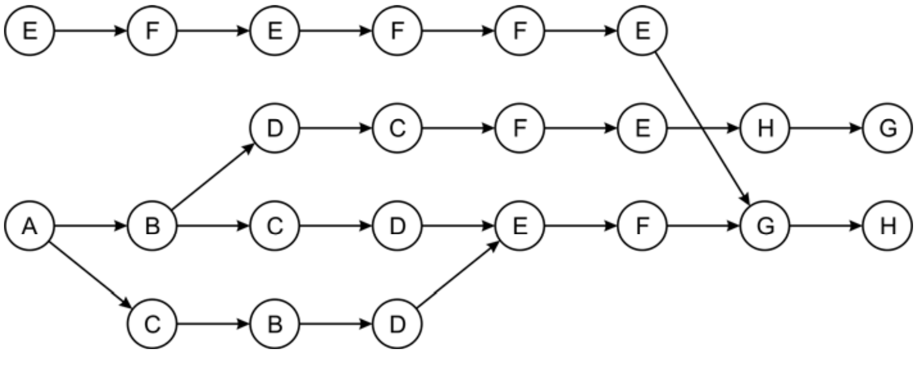
· the out-degree of the last vertex is 0

In other words, paths can fork only on the first K letters, and they can meet only on the last K letters.

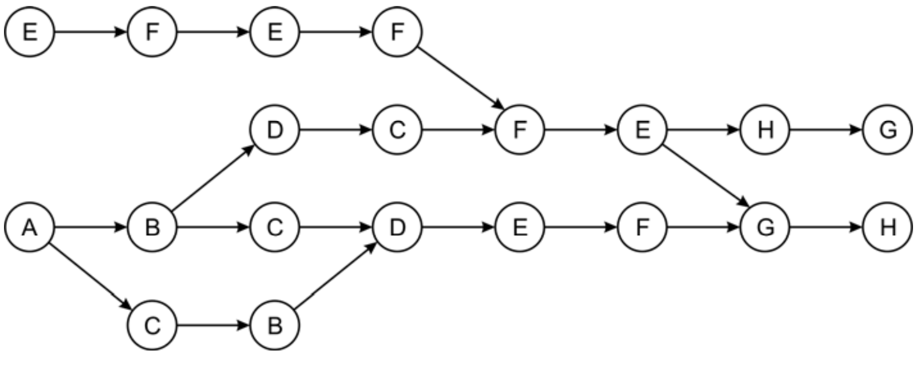
For the given set of the words, we say that the "kokos" is **minimal** if the total number of vertices is **as small as possible**.

Write a program that will find **the number of vertices** in a minimal kokos.

An example of a minimal kokos (the set of the words is from the third example):



It may seem that we can compact the graph like this:



However, this graph is not a **kokos** because paths meet on the 4th letter (D), and they fork on the 6th letter (E).

input data

The first line of input contains two integers N and K, 1 ≤ N ≤ 10 000, 1 ≤ K ≤ 100.

Each of the following N lines contains one word from the set. All letters will be uppercase letters of the English alphabet ('A'-'Z').

output data

The first and only line of output should contain the number of vertices in a minimal kokos.

examples

input

```
2 4
ABCDEFGH
EFGHIJKL
```

output

```
16
```

input

```
4 3
XXZZXX
XXYYZZ
AABBCZ
ABCZZZ
```

output

```
18
```

input

```
4 4
ABCDEFGH
ACBDEFGH
ABDCFEHG
EFEFFEGH
```

output

```
23
```