Covel Playlist

Problem ID: covelplaylist

Mouthwatering Cuisine at Covel

The music playlist at Covel (now dubbed Epicuria) offers instrumental tracks that students just can't get enough of. The songs consist of N piano notes (A-G) and they are always so catchy. As a result, students decide to use the app Shazam to figure out what songs to add to their playlists, but there seems to be a problem: the app can never figure out what the song is.

You start up Shazam at some note of the song currently playing, at which point the app listens to K notes and scavenges its database for a match (i.e. a song in the database containing the same sequence of K notes somewhere in the song). If the end of the song is reached before Shazam has listened to K notes, it will instead only match songs in the database that *end* with the same sequence it heard. Shazam must always listen to at least one note (i.e. K must be a positive integer).

Shazam definitely has the currently playing song in its database, but if there are other songs in the database that match the sequence heard, it will fail to identify the song. While it would be nice to listen until the end of the song each time, Shazam is constrained by its resources and it wants to find an optimal value of K, given that users could start listening at any note of the song. Given the currently playing song, as well as Shazam's song database, determine the minimum value of K such that at least 50% of the starting positions result in Shazam identifying the song.

Input

The input starts with one line containing two space-separated integers N and M. These denote the length of each song ($1 \leq N \leq 1\,000$) and the number of songs in the database ($1 \leq M \leq 1\,000$).

The following M lines represent the songs in the database. Each line is a string of N characters ($A - G$), giving the notes of the song. The first of these M lines is the song currently playing. It is guaranteed that no two songs are exactly the same.

Output

Output one line containing the minimum possible value of K such that 50% of starting positions of the song currently playing allow Shazam to find a match, or -1 if there is no such K.

Sample Explanation

In the first sample case, if Shazam only listens to two notes, it can be started from the 1st, 2nd, or 3rd note to detect unique note combinations, but not the 4th position (as then the third song would match too). This gives 3 valid starting notes out of 4 which is greater than 50%.

In the second sample case, even if Shazam listens to all 4 notes, there is only one starting note that Shazam can identify the song from, which is when it is started from the first note and it listens to $AAAA$. Starting from the second, third, or fourth starting notes cause Shazam to listen to $AAA*$, $AA*$, and $A*$ respectively, where $*$ represents the end of the song. All of these also match with the third song in the database ($BAAA$), so none of the other starting positions provide a unique match.

Sample Input 1	Sample Output 1
4 3 ABAB AAAA BBBB	2

Sample Input 2	Sample Output 2
4 3 AAAA AAAB BAAA	-1