Task Description: Finding Words

You subscribed to a local newspaper that has a weekly puzzle. In the puzzle, you are given a R*C matrix with each cell consist of a single lowercase alphabet. In the matrix there are several words which the characters are constructed :

- Horizontally
  - from left to the right, or
  - from right to the left
- Vertically
  - from top to the bottom, or
  - from bottom to the top
- Diagonally
  - from top-left to the bottom-right, or
  - from top-right to the bottom-left, or
  - from bottom-left to the top-right, or
  - from bottom-right to the top-left

In the puzzle you are also given N words. You have to determine whether each word is exist in the matrix.
You think that this puzzle is too hard for you, especially when the size of the matrix is big. Therefore, you decide to write a program that can helps you.

Input

The first line will consist of two integers separated by single space representing R and C respectively.
The next R lines consist of C characters representing the matrix.
The next line will consist of a single integer representing N.
The next N lines consist of a string that represents a query.

Output

For each query, you have to output "YES" (without quote) if the string query is exist in the matrix, or "NO" (without quote) otherwise.

Sample Input

```
7 8
xassddqs
gzccdsxu
jonathan
dazzzzzt
wplofeee
cadddemc
aweqaedk
6
jonathan
kde
gnu
gocode
sunteck
cstentwenty
```

Sample Output

```
YES
```

```
YES
NO
YES
YES
NO
```

Constraint

N will be between 1 and 100 inclusive.

R and C will be between 1 and 100 inclusive.

Each word in the query will consist of only lowercase alphabet and will be between 1 and 100 characters inclusive.

Skeleton program

```
/**
*     Name              :
*     Matric-number     :
*     Plab account      :
*/
#include <iostream>
#include <algorithm>
#include <cstring>
#include <cmath>
#include <cstdio>
#include <string>
using namespace std;

/**
*     Check whether a string is exist in a matrix
*     Pre-condition  :
*     Post-condition :
*/
int solve(int R, int C, string matrix[100], string query) {

      return 0;
}

int main(){
    // read input

    // process the input

    // output
    return 0;
}
```

Note:

1. You should develop your program in the subdirectory, ex1 and use the cpp file provided. You should not create new file or rename the file provided.

2. You don't have to use OOP in this sit in lab. You are allowed to add more methods inside each file.
3. If your algorithm is different from the given skeleton, you can develop according to your own algorithm.
4. Please be reminded that the marking scheme is
   Input:10%, Output:10%, Programming Style:30% and Correctness: 50%