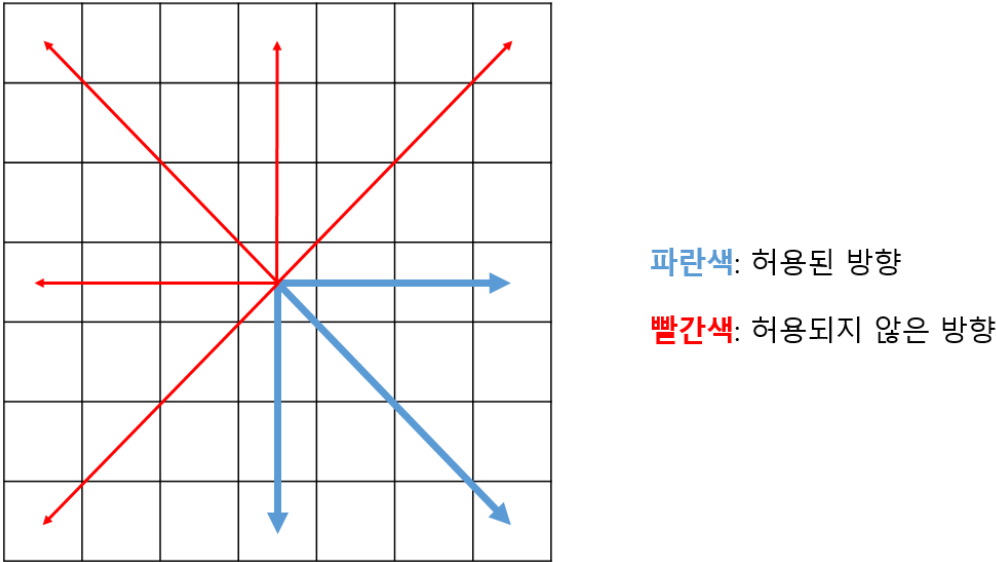


단어 퍼즐

단어 퍼즐은 $N \times N$ 크기의 격자판에서 주어진 단어들이 존재하는지 확인하는 퍼즐이다. 각 단어들은 격자판에서 상하, 좌우, 대각선으로 문자들이 연속해서 존재해야 한다. 이때 단어들은 상→하, 좌→우, 좌상단→우하단 방향으로 읽는 것만 허용되며, 다른 방향으로 읽는 것은 허용되지 않는다. [그림 1]은 격자판에서 단어를 읽는 8 가지 방향 중 허용되는 방향은 파란색, 허용되지 않는 방향은 빨간색으로 표시한 예이다. 한편, 단어들은 반드시 한 방향으로만 읽어야 하며, "ㄱ"자 와 같이 방향을 변경하는 것은 허용되지 않는다. 이때 단어들이 허용되는 방향으로 존재한 경우에만, 해당 단어가 존재한다고 한다. 격자판과 단어들이 주어졌을 때, 각 단어가 격자판에 존재하는지 확인하는 프로그램을 작성하시오.

예를 들어, [그림 2]은 한 격자판과 "APPLE", "PIANO" "PIG"가 주어졌을 때, 모든 단어들이 존재하는 예시를 보여 준다. [그림 3]은 [그림 2]의 격자판과 "FIGHT", "EGG", "HELLO", "COIN"이 주어졌을 때, 모든 단어들이 존재하지 않는 예시를 보여준다.



[그림 1] 격자판에서 단어를 읽는 허용된 방향과 허용되지 않은 방향

A	P	P	L	E	J	F
B	I	O	I	K	I	A
M	A	T	W	G	R	E
R	N	A	H	E	G	P
C	O	T	P	Q	Y	E
Z	I	O	L	L	E	H
C	N	I	N	E	X	U

[그림 2] 허용되는 읽기 방법 (성공)

A	P	P	L	E	J	F
B	I	O	I	K	I	A
M	A	T	W	G	R	E
R	N	A	H	E	G	P
C	O	T	P	Q	Y	E
Z	I	O	L	L	E	H
C	N	I	N	E	X	U

[그림 3] 허용되지 않는 읽기방법 (실패)

※ 프로그램의 실행 시간은 8 초, 메모리 사용량은 512MB 를 초과할 수 없다.

사용할 수 있는 언어는 C, C++로 제한한다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin, cout 의 입출력 속도를 개선할 수 있다.

```
ios_base::sync_with_stdio(false);  
cin.tie(NULL);  
cout.tie(NULL);
```

단, 위의 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력을 혼용해서 사용하면 안된다. C++의 std::endl 의 경우 출력 속도가 느리므로, cout<<endl; 대신 cout<<"\n";을 사용하는 것을 권장한다.

입력

첫 번째 줄에 격자판의 크기 N ($5 \leq N \leq 3,000$)과, 단어들의 개수 M ($1 \leq M \leq 50$)이 공백으로 구분되어 주어진다.

두 번째 줄부터 N 개의 줄에 걸쳐, 각 줄마다 N 개의 문자들이 주어진다.

이후 M 개의 줄에 걸쳐, 각 줄마다 길이가 S ($1 \leq |S| \leq N$)인 단어들이 주어진다.

단, 격자판과 단어들은 모두 알파벳 대문자로 구성되어 있다.

출력

M 개의 줄에 걸쳐, 각 단어가 격자판에 존재한다면 1, 아니라면 0 을 한 줄에 하나씩 출력한다.

예제 입출력

예제 입력	예제 출력
7 12	1
APPLEJF	0
BIOIKIA	1
MATWGRE	0
RNAHEGP	0
COTPQYE	0
ZIOLLEH	0
CNINEXU	1
PIG	1
SOLVE	0
APPLE	1
HELLO	0
EGG	
OPERA	
COIN	
PIANO	
POTATO	
HOPE	
NINE	
YEAR	

예제 입력	예제 출력
6 2	1
ABCABA	0
AACABA	
AABAAB	
BCCAAC	
ABBCAA	
ACABBC	
ABAAB	
AAACA	