




시험 준비(문장 맞추기)

컴퓨터공학과 N 명의 친구들은 인하대학교에서 어렵기로 소문난 고급 영어 강의를 수강하고 있다. 이 수업에서 교수님은 교과서에 있던 수많은 문장들 중 시험에 출제할 문장들을 말씀해 주셨다. 그 문장들만 다 외운다면 시험에서 좋은 점수를 받을 수 있을 것이다. 다만 그 교수님의 말씀이 너무 빨라서, 강의시간에 문장들 전체를 기억할 수는 없었다. 친구들은 각자 교수님께서 말씀하시는 영어 단어들 중 일부만을 노트에 적을 수 있었다. 수업이 끝난 후에는 친구들이 모두 모여 노트에 필기한 내용을 바탕으로 교수님께서 말씀하신 문장을 찾아내려고 한다. 확인해보니 놀랍게도, 각자 필기한 단어들은 모두 달랐다. 컴퓨터공학과와 N 명의 친구들이 각자 적은 문장들은 아래와 같은 특성이 있다.




- ① 모든 친구들은 문장의 일부 단어들을 교수님이 말씀하신 문장의 순서를 지키며 적을 수 있었다. 단, 교수님이 말씀하지 않은 단어를 적거나, 말씀하신 단어를 잘못 적은 경우는 없다.
- ② 친구들은 팀웍이 매우 좋아서 모든 친구들이 서로 다른 단어를 적었고, 교수님이 말씀하신 문장의 모든 단어를 친구 중 누군가는 반드시 적을 수 있었다. 즉, 교수님이 말씀하신 문장의 모든 단어는 친구 중 한 명이 반드시 적었다.

특정한 문장이 주어졌을 때, 교수님께서 처음으로 말씀하신 문장일 수 있는지 여부를 판단하는 프로그램을 작성하시오.

예를 들어 3 명의 친구들이 작성한 필기 내용이 아래 [그림 1]과 같다고 하자. 주어진 문장 “use binary search to solve the problem”에 대해서 교수님이 처음 말씀하신 문장일 수 있는지 판단한다. 그렇다면 “use” 라는 단어는 3 명 중 한명이 반드시 적었을 것이며, 그 학생은 “use”라는 단어를 다른 어떤 단어보다 먼저 적었을 것이다. 그다음 나온 “binary” 역시 누군가의 필기한 내용에 있을 것이며, 그 학생은 “use”를 제외한 나머지 단어중 가장 빠른 순서에 필기가 되어 있을 것이다. 이와 같이 주어진 문장에 나온 모든 단어들이 순서에 맞게 필기가 되어 있다면 그 문장은 필기하고자 했던 단어가 맞다고 판단한다. 결국 [그림 2]와 같이 “use binary search to solve the problem” 이라는 문장을 듣고 필기한 내용이 맞다고 판단한다. 그러나, “use binary tree to solve the problem” 이라는 문장이 주어졌다면, tree 라는 단어가 필기 되지 않았으므로, 교수님이 말씀하신 문장일 수 없다. “ordered array to use binary search” 라는 문장이 주어졌다면, ordered 라는 단어를 처음 적은 학생이 없으므로 교수님이 처음으로 말씀하신 문장일 수 없다.

친구 1		use	search	solve	study
친구 2		binary	the	problem	
친구 3		to	ordered	array	hard

[그림 1] $N = 3$ 일 때, 필기 내용의 예시

친구 1		1 use	3 search	5 solve	study
친구 2		2 binary	6 the	7 problem	
친구 3		4 to	ordered	array	hard

[그림 2] “use binary search to solve the problem”문장에 대해 3 명이 필기한 순서

※ 프로그램의 실행 시간은 2 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 테스트 케이스 수 T ($1 \leq T \leq 1,000$)가 주어진다.

각 테스트 케이스의 구성은 다음과 같다.

- 첫 번째 줄에 친구들의 수 N ($1 \leq N \leq 20$)과 교수님께서 처음 말씀하신 문장일 수 있는지 판단할 문장을 구성하는 단어의 수 M ($1 \leq M \leq 2,000$)이 공백으로 구분되어 주어진다.
- 두 번째 줄에 교수님께서 처음 말씀하신 문장일 수 있는지 판단할 문장을 구성하는 M 개의 단어 w_i ($1 \leq i \leq M, 1 \leq |w_i| \leq 7$)가 공백으로 구분되어 주어진다.
- 이후 N 개의 줄에 걸쳐 각 친구들이 노트에 필기한 단어의 수 K_j ($1 \leq j \leq N, 1 \leq K_j \leq 100$)와 K_j 개의 단어 $s_{j,k}$ ($1 \leq j \leq N, 1 \leq k \leq K_j, 1 \leq |s_{j,k}| \leq 7$)들이 공백으로 구분되어 주어진다.
- 모든 단어들은 알파벳 소문자로 구성되며, 노트에 작성된 단어들은 모두 다르게 주어진다.

출력

각 테스트 케이스마다 필기한 내용을 이용하여 규칙에 따라 교수님께서 처음으로 말씀하신 문장일 수 있으면 1, 없으면 0을 한 줄에 하나씩 출력한다.

예제 입출력

예제 입력	예제 출력
3	1
3 7	0
use binary search to solve the problem	1
4 use search solve study	
3 binary the problem	
4 to ordered array hard	
3 6	
ordered array to use binary search	
4 use search solve study	
3 binary the problem	
4 to ordered array hard	
3 6	
ordered array to use binary search	
5 to use search solve study	
3 binary the problem	
3 ordered array hard	