



2. Play Onwards

2년이 지난 아직도 서훈이는 용훈이의 아재개그에 고통받고 있다. 이번에 용훈이가 새로 구사하는 아재개그는 반댓말을 이용한 아재개그이다.

"프로그램의 반댓말은 뭐게? 비기너톤! ㅋㅋㅋㅋㅋㅋ"

용훈이의 아재개그에 충격먹은 서훈이는 용훈이의 모든 아재개그를 기록하기 시작하였다. 그러자 서훈이는 용훈이의 아재개그에 있는 규칙을 발견할 수 있었다.

1. 모든 단어는 알파벳 소문자만으로 표현된다.
2. 서훈이가 M 글자 단어를 말하면 용훈이가 M 글자 단어로 대답한다.
3. 용훈이는 N 개의 M 글자 단어가 들어있는 사전에서 단어를 골라 대답한다.
4. 용훈이는 사전에서 서훈이가 말한 단어와 가장 다른 단어를 고른다. 가장 다른 단어란 두 단어의 첫 번째 글자, 두 번째 글자, ..., M 번째 글자끼리 비교했을 때 서로 다른 글자 쌍의 수가 최대가 되는 단어를 의미한다.
5. 용훈이가 고를 수 있는 단어가 여러 개라면 그중 무작위로 하나를 선택한다.

이제 서훈이는 요령이 생겨서 용훈이의 대답만 듣고 자신이 무슨 말을 했었는지 알 수 있다. 당신도 할 수 있다! 용훈이에게 말을 거는 프로그램과 용훈이의 대답을 듣고 자신이 말했던 단어를 유추하는 프로그램을 작성해보자.

요구사항

당신은 파일 두 개를 제출해야 한다.

첫 번째 파일은 `make.cpp`이다. 이 파일에서 당신은 다음 함수를 구현해야 한다. 이 파일은 `make.h`와 같이 컴파일된다.

```
string MakeWord(int N, int M, string[] Dict)
```

- N : 사전에 있는 단어의 수.
- M : 단어의 길이.
- $Dict$: 크기 N 짜리 배열. $0 \leq i \leq N - 1$ 를 만족하는 i 에 대해서, $Dict[i]$ 는 사전의 i 번째 단어이다. 사전의 모든 단어는 서로 다르다.
- 이 함수는 한 번 호출된다.
- 이 함수는 길이 M 짜리 문자열을 반환해야 한다. 해당 문자열은 용훈이에게 말할 단어를 의미한다. 이 문자열은 알파벳 소문자로만 이루어져야 한다.

두 번째 파일은 `guess.cpp`이다. 이 파일에서 당신은 다음 함수를 구현해야 한다. 이 파일은 `guess.h`

와 같이 컴파일된다.

```
string GuessWord(int M, string YH)
```

- M : 단어의 길이.
- YH : 길이 M 짜리 문자열. 이 문자열은 용훈이의 대답을 의미한다.
- 이 함수는 한 번 호출된다.
- 이 함수는 길이 M 짜리 문자열을 반환해야 한다. 해당 문자열은 앞서 자신이 말했던 단어를 의미한다.

위에 언급된 조건을 만족하지 않거나 단어를 유추하는 데 실패했다면, 당신의 프로그램은 **Wrong Answer**으로 채점된다. 나머지 경우에는 당신의 프로그램은 **Accepted**으로 채점된다.

중요 공지사항

- 실제 채점 시에는 제출한 두 프로그램이 독립적으로 컴파일 및 실행된다.
- 시간 및 메모리 사용량은 각 프로그램의 사용량 합으로 계산된다.
- 작성한 프로그램에서 표준 입력과 표준 출력을 사용하면 안 되고, 어떠한 파일에도 접근하면 안 된다. 이를 어길 시 당신의 프로그램이 **Wrong Answer**로 채점될 수 있다.

예제

다음 호출을 고려해보자.

```
MakeWord(4, 7, ["weekend", "evening", "chicken", "alcohol"])
```

이 함수가 "cafeine"을 반환한다면 guess.cpp에서 아래와 같은 호출이 발생한다.

```
GuessWord(7, "alcohol")
```

이 함수는 반드시 "cafeine"을 반환해야 한다.

제한

- $1 \leq N \leq 100$
- $1 \leq M \leq 10$
- $|Dict[i]| = M$ ($0 \leq i \leq N - 1$)
- $Dict[i]$ ($0 \leq i \leq N - 1$)는 알파벳 소문자로만 이루어져 있다.
- $Dict[i] \neq Dict[j]$ ($0 \leq i < j \leq N - 1$)

부분문제

1. (100점) 추가 제약조건은 없다.

이 문제는 다른 문제들과는 다른 방식으로 채점된다. 전체 테스트 케이스의 수를 T , Accepted로 채점된 테스트 케이스의 수를 t 라 했을 때, P 점을 받는다.

- $P = \lfloor 100 \times t/T \rfloor$.

샘플 인터페이스

문제 페이지에서 샘플 코드를 다운로드받을 수 있다. 만약 Visual Studio나 Eclipse, Code::Blocks와 같은 IDE 툴을 사용한다면 `make.cpp`, `make.h`, `guess.cpp`, `guess.h`, `grader.cpp`를 한 프로젝트에 넣어서 컴파일하면 된다.

터미널에서 코드를 컴파일한다면 아래 컴파일 명령어를 이용하면 된다.

```
g++-7 -Wall -lm -static -DEVAL -o onwards -O2 grader.cpp make.cpp guess.cpp -std=c++17
```

답안을 제출할 때에는 `make.cpp`와 `guess.cpp`를 제출하면 된다.

Input format

- line 1: $N M$
- line $2 + i$ (모든 $0 \leq i \leq N - 1$ 에 대해): $Dict[i]$

Output format

당신의 프로그램이 Accepted으로 채점되었다면, 샘플 그레이더는 첫 번째 줄에 **Correct**를 출력한다.

당신의 프로그램이 **Wrong Answer**으로 채점되었다면, 샘플 그레이더는 첫 번째 줄에 에러 메시지를 출력한다.