

EMP

적군의 EMP(Electromagnetic Pulse) 공격으로 데이터센터의 파일들이 손상되었다. 각 파일은 하나 이상의 문자로 구성되어 있으며, 각 문자는 비트열로 표현된다. 김인하 박사는 파일의 변경된 문자들을 원래대로 복원하기 위해, 변경된 문자들을 분석하였다. 분석 결과, 문자들은 비트열의 일부분만 다른 비트로 변경되었으며, 해밍 거리(Hamming distance)를 이용하여 복원할 수 있다는 것을 발견하였다. 길이가 같은 두 비트열의 해밍 거리는 동일한 위치의 비트 값이 불일치한 위치들의 개수이다. 즉, 변경된 문자들은 해밍 거리가 최소인 문자로 복원할 수 있으며, 해밍 거리가 최소인 문자는 항상 유일했다.

예를 들어, 알파벳 {A, B}만 고려할 때 문자 A가 00000000, 문자 B가 11111111이라면, 변경된 문자의 비트열 00001000은 해밍 거리가 1로 최소인 문자 A로 복원할 수 있다. 00001111과 같이, 두 개 이상의 문자와 해밍 거리가 최소인 경우(A, B 모두 해밍 거리가 4)는 발생하지 않았다.

알파벳의 개수, 문자에 대한 비트열의 길이, 각 문자의 비트열, 그리고 손상된 파일이 주어질 때, 파일의 변경된 문자들을 원래 문자들로 복원하는 프로그램을 작성하시오.

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

이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 첫 번째 줄에는 알파벳을 구성하는 문자 개수 M ($1 \leq M \leq 26$), 문자에 대한 비트 수 N ($1 \leq N \leq 128$)이 공백으로 구분되어 주어진다.
- 두 번째 줄부터 M 개의 줄에는 각 문자의 비트열이 한 줄에 하나씩 주어진다. 단, 주어진 문자의 비트열들은 차례대로 영문 대문자 A, B, C, ... 에 해당한다고 간주한다.
- 마지막 줄에는 파일에 대한 비트열의 길이 L ($1 \leq L \leq 1,024$) 과 실제 비트열이 공백으로 구분되어 주어진다. 파일에 대한 비트열의 길이는 N 의 배수이다.

Handwritten calculation showing complexity analysis: $1000 \times 8 \times 128 \times 128$, $O(n^2)$, and $128000,000$.

출력

각 테스트 케이스마다 파일의 문자들을 복원한 문자열을 한 줄에 출력한다.

예제 입출력

예제 입력	예제 출력
2 3 8 00000000 11111111 01010101 24 010000001111110101011101 4 8 00000011 00001100 00110000 11000000 16 1000000000000001	ABC DA