

문제	2-A
제목	<b>오류 정정</b>
내용	<p>오류 정정(error correction)이란, 비트로 표현된 데이터의 저장 또는 전송 중 일부가 손상된 경우 이를 원래의 데이터로 복원하는 기술이다. 예를 들어 A를 나타내는 코드가 00000000, B를 나타내는 코드가 11111111이라면, 00001000은 원래는 A였던 데이터의 한 비트에 오류가 생긴 것으로 간주하는 것이 더 타당하므로, A로 해석할 수 있다. A, B와 같은 데이터를 00000000, 11111111과 같은 코드로 바꾸는 과정을 인코딩(encoding), 그 반대 과정을 디코딩(decoding)이라 하며, 00000000과 같이 한 문자에 대한 인코딩 결과를 코드워드(codeword)라 한다. 길이가 같은 두 비트열에 대해 비트값이 다른 위치의 개수를 해밍 거리(Hamming distance)라고 정의할 때, 오류 정정 알고리즘은 주어진 비트열로부터 해밍 거리가 가장 작은 코드워드로 복원을 시도하게 된다. (단, 모든 코드워드는 길이가 같다.) 그런데 00001111과 같이 오류가 너무 많은 경우는 A로 복원해야 할지 B로 복원해야 할지 결정하는 것이 어려우며(둘 다 해밍 거리가 4임), 만약 5비트 이상의 오류가 생긴 경우는 복원이 잘못될 수도 있다. 예를 들어 A에 5비트 오류가 생겨 00011111이 된 경우, 복원 알고리즘은 이 비트열을 해밍 거리가 더 작은 B로 잘못 복원할 것이다. 알파벳의 개수, 코드워드의 길이, 각 알파벳에 대한 코드값(인코딩 규칙), 그리고 인코딩된 비트열이 주어질 때, 오류를 정정하여 원래 텍스트를 복원하는 프로그램을 작성하시오. 단, 오류는 비트를 다른 비트로 바꾸는 식으로만 발생되며, 비트가 소실되는 경우(예: 00000000 → 0000000)는 없다고 가정한다. 또한, 오류가 너무 많이 발생하여 오류 정정 알고리즘이 복원에 대한 결정을 할 수 없거나(두 개 이상의 코드워드와의 해밍거리가 같은 경우) 복원에 실패하는 경우는 없다고 가정한다. 즉, 오류는 최소 해밍 거리 우선 방식의 오류 정정 결과가 맞게 되는 범위 안에서만 발생한다.</p> <p>사용할 수 있는 언어는 C, C++로 제한한다. 프로그램의 실행 제한 시간은 1초이다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin 입력 속도를 개선할 수 있다.</p> <pre>std::ios::sync_with_stdio(false);</pre>
입력 형식	<p>입력은 표준입력으로 다음과 같이 주어진다. 첫 줄에는 테스트케이스의 수 <math>T(\leq 1000)</math>가 주어진다. 각 테스트케이스의 첫 번째 줄에는 알파벳의 개수 <math>M(\leq 26)</math>, 각 코드워드의 비트 수 <math>N(\leq 128)</math>이 공백 하나를 사이에 두고 주어진다. 각 테스트케이스의 두 번째 줄부터 <math>M</math>개의 줄에는 각 알파벳의 코드워드가 한 줄에 하나씩 주어진다. 단, 주어진 코드워드들은 차례대로 영문 대문자 A, B, C, ... 에 해당한다고 간주한다. 각 테스트케이스의 마지막 줄에는 인코딩된 비트열의 길이(<math>\leq 1024</math>)와 실제 비트열이 공백 하나를 사이에 두고 주어진다. 이후에 같은 형태의 테스트케이스가 <math>T-1</math>개 더 반복된다.</p>
출력 형식	<p>출력은 표준출력으로 수행하며 총 <math>T</math>줄로 이루어진다. 각 테스트케이스별로 오류가 정정된 문자열을 한 줄에 출력한다.</p>

예	입력	2 3 8 00000000 // A의 코드워드 11111111 // B의 코드워드 01010101 // C의 코드워드 24 010000001111110101011101 4 8 00000011 // A의 코드워드 00001100 // B의 코드워드 00110000 // C의 코드워드 11000000 // D의 코드워드 16 1000000000000001 (empty line)
	출력	ABC DA (empty line)