

KMP 알고리즘 (with fail function)

주어진 문자열에서 특정한 패턴 문자열을 검색하는 문자열 검색 알고리즘 중에서 *fail()* 함수를 이용한 KMP(Knuth-Morris-Pratt) 알고리즘을 구현하시오.

이 알고리즘에서 핵심은 패턴 문자열에 해당하는 *fail()* 함수를 만드는 것이다. 패턴 문자열 `pattern[]`의 길이를 n 이라고 할 때, 함수 *fail(k)*, $0 \leq k < n$, 는 패턴 문자열의 prefix인 `pattern[0] ... pattern[k]`의 최대 중첩 길이, 즉 해당 prefix의 prefix부분과 일치하는 가장 긴 suffix의 길이를 나타낸다.

예를 들어 길이가 8인 패턴 문자열 **ABABABAC**에 해당하는 *fail(k)*의 값은 다음과 같다.

k	<i>prefix</i>	<i>fail(k)</i>
0	A B A B A B A C	0
1	A B A B A B A C	0
2	A B <u>A</u> B A B A C	1
3	A B <u>A</u> B <u>A</u> B A C	2
4	A B <u>A</u> B <u>A</u> B <u>A</u> C	3
5	A B <u>A</u> B <u>A</u> B <u>A</u> C	4
6	A B <u>A</u> B <u>A</u> B <u>A</u> C	5
7	A B A B A B A C	0

검색하고자 하는 패턴 문자열과 검색범위를 나타내는 텍스트 문자열이 주어졌을 때, 패턴 문자열의 *fail()* 함수를 만들고, 텍스트 문자열에 패턴 문자열이 몇 번 나타나는지를 계산하는 프로그램을 작성하시오.

입력

입력은 표준입력(standard input)을 사용한다. 입력은 t 개의 테스트 케이스로 주어진다. 입력 파일의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수 t 가 주어진다. 두 번째 줄부터 한 줄에 한 개의 테스트 케이스가 입력된다. 각 테스트 케이스에 해당되는 각 줄에는 패턴 문자열과 텍스트 문자열이 주어진다. 두 문자열의 최소 길이는 1이며 최대 길이는 1000이다. 두 문자열은 모두 영문자 대문자 **A**, **B**, **C**로만 만들어 진다고 가정한다. 두 스트링 사이에는 한 개의 공백이 있으며, 잘못된 데이터가 입력되는 경우는 없다.

출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트 케이스의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 줄에 입력되는 길이가 n 인 패턴문자열에 해당하는 *fail()* 함수를 만들었을 때, *fail(0)* 부터 *fail(n-1)* 까지 순서대로 출력한다. 각 정수들 사이에는 한 개의 공백을 둔다. 두 번째 줄에는 텍스트 문자열에 패턴 문자열이 몇 번 나타나는지를 출력한다.

입력과 출력의 예

입력
3 ABABABAC ABCABAABABABACACA AAAAA AAAAAAAAAA ABCABCABC CBACBACBACBACBA

출력
0 0 1 2 3 4 5 0 1 0 1 2 3 4 6 0 0 0 1 2 3 4 5 6 0