

부품 재조립

시간 제한	1초
메모리 제한	512 MB
사용 가능 언어	C, C++
C++ 허용 STL	전체
C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가하여 cin, cout 의 입출력 속도를 개선할 수 있다. <div>ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);</div> 단, 위 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력과 혼용해서 사용하면 안 된다. C++ std::endl의 경우 출력 속도가 느리므로, cout<<endl; 대신 cout<<"\n"; 을 사용하는 것을 권장한다.	

인덕 공장은 여러 가지 부품을 조립하여 하나의 제품을 만들어낸다. 부품은 모두 일렬로 조립하며, 각 부품은 영어 알파벳 대문자로 주어진다.

인덕 공장에서 일하고 있는 인하는 필요한 부품을 빠트리거나, 불필요한 부품을 추가하거나, 다른 종류의 부품을 조립하는 실수를 자주 저지른다. 사수 비룡이는 인하가 공장장에게 혼나지 않게 하기 위해 인하가 잘못 조립한 제품을 설계에 맞게 보수하려 한다. 보수하는 방법은 세 가지가 있다.

- 1. 제품의 특정 위치에 부품 하나를 삽입한다.
- 2. 제품의 특정 위치에 있는 부품 하나를 제거한다.
- 3. 제품의 특정 위치에 있는 부품 하나를 다른 종류의 부품으로 교체한다.

제품을 한 번 보수할 때마다 비용이 1만큼 필요하다.

예를 들어, 인하가 조립한 제품이 ABB이고, 원래 만들어야 하는 제품이 ACBA라고 하자. 이 경우 두 번째 위치에 있는 부품을 B에서 C로 바꾸고, 마지막 위치에 부품 A를 삽입하면 올바른 제품 ACBA로 만들 수 있다. 이 경우 보수 비용은 2이며, 이때가 보수 비용의 최솟값이다.

다른 예로, 인하가 조립한 제품이 ABBCD이고, 원래 만들어야 하는 제품이 ABCDE라고 하자. 만약 비룡이가 세 번째 위치에 있는 부품을 B에서 C로 바꾸고, 네 번째 위치에 있는 부품을 C에서 D로 바꾸고, 마지막 위치에 있는 부품 D를 E로 바꾸면, 보수 비용은 3이 든다. 한편, 두 번째 위치에 있는 부품을 제거하고, 마지막 위치에 부품 E를 삽입하면, 보수 비용은 2이고 이때가 보수 비용의 최솟값이다.

비룡이가 올바른 제품을 만들기 위한 **보수 비용**의 최솟값을 구하여라.

입력

첫 번째 줄에는 테스트 케이스 수 T ($1 \leq T \leq 30$)가 주어진다. 이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 첫 번째 줄에 인하가 조립한 제품 S_1 ($1 \leq |S_1| \leq 3,000$)이 주어진다.
- 두 번째 줄에 원래 만들어야 하는 제품 S_2 ($1 \leq |S_2| \leq 3,000$)가 주어진다.

단, 제품의 부품들은 영어 알파벳 대문자로 주어진다.

출력

각 테스트 케이스마다 보수 비용의 최솟값을 한 줄씩 출력한다.

예제 입출력

예제 입력	예제 출력
3 ABB ACBA ABB ABB ABBCD ABCDE	2 0 2