

## 문제1 : 문자열 찾기

영어 소문자로 구성된 두 문자열  $A$ 와  $B$ 에 대해서 다음의 조건이 만족될 때, 두 문자열은 “사실상 같다”고 한다.

1.  $A$ 와  $B$ 의 길이가 같다.
2. 모든 가능한 정수  $i$ 와  $j$ 에 대해  $A$ 의  $i$ 번째와  $j$ 번째 글자가 같으면  $B$ 의  $i$ 번째와  $j$ 번째 글자도 같다.
3. 모든 가능한 정수  $i$ 와  $j$ 에 대해  $A$ 의  $i$ 번째와  $j$ 번째 글자가 다르면  $B$ 의  $i$ 번째와  $j$ 번째 글자도 다르다.

예를 들어,  $A = aba$ 와  $B = pqp$ 는 사실상 같은 문자열들이다. 하지만,  $A = abca$ 와  $B = abcb$ 는 사실상 같은 문자열의 쌍이 아니다.

문자열  $T$ 와  $P$ 를 받아서  $T$ 의 연속된 부분문자열들 중  $P$ 와 사실상 같은 부분문자열의 개수를 계산하는 프로그램을 작성하라.

예를 들어,  $T = abababbab$ 이고  $P = pqp$ 인 경우  $T$ 의 왼쪽부터  $aba$ ,  $bab$ ,  $aba$ ,  $bab$ ,  $bab$ 의 5개의 부분문자열이  $P$ 와 사실상 같은 것들을 알 수 있다.

여러분은 다음 함수를 작성하여야 한다.

- `int findP( char T [], char P [], int N, int M )` :  $T$ 는 길이  $N+1$ 인 배열(문자열)이다.  $P$ 는 길이  $M+1$ 인 배열이다.  $T$ 와  $P$ 에는 각각 길이  $N$ 과  $M$ 인 영어 소문자 문자열이 저장되어 있다.  $T$ 와  $P$ 의 마지막 위치에는 `'\0'`이 저장되어 있다. `findP`는  $T$ 의 연속된 부분문자열들 중  $P$ 와 사실상 같은 부분문자열의 개수를 리턴해야 한다.

### 구현 세부사항

여러분은 `match.cpp`라는 이름의 정확히 하나의 파일을 제출해야 한다. 이 파일에는 다음의 함수가 구현되어 있어야 한다.

- `int findP( char T [], char P [], int N, int M )`

이 함수는 위에서 설명한 것과 같이 동작하여야 한다. 물론 다른 함수들을 만들어서 내부적으로 사용할 수 있다. 제출한 코드는 입출력을 수행하거나 다른 파일에 접근하여서는 안된다.

### grader 예시

주어지는 grader는 다음과 같은 형식으로 입력을 받는다.

- line 1: 문자열  $T$
- line 2: 문자열  $P$

주어진 grader는 여러분의 코드가 `findP()` 함수에서 리턴한 값을 한 줄에 출력한다.

제한 조건

- $1 \leq N \leq 1,000,000, 1 \leq M \leq N.$

서브태스크 1 [ 8 points]

- $N = M$

서브태스크 2 [ 5 points]

- $1 \leq N \leq 100$

서브태스크 3 [ 12 points]

- $1 \leq N \leq 2,000$

서브태스크 4 [ 75 points]

- 원래 제한 조건 이외의 추가적인 제한 조건이 없음

[입력 예]

abababbab
pqp

위의 입력에서 여러분의 코드가 동작하는 방식에 따른 실행 예를 아래에 보인다.

호출	리턴 값
<code>findP( "abababbab" , "pqp" , 9 , 3 )</code>	5