**문제) 문자열 내 p와 y의 개수**

**문제 설명**

대문자와 소문자가 섞여있는 문자열 s가 주어집니다. s에 'p'의 개수와 'y'의 개수를 비교해 같으면 True, 다르면 False를 return 하는 solution를 완성하세요. 'p', 'y' 모두 하나도 없는 경우는 항상 True를 리턴합니다. 단, 개수를 비교할 때 대문자와 소문자는 구별하지 않습니다.

예를 들어 s가 pPoooyY면 true를 return하고 Pyy라면 false를 return합니다.

**제한사항**

문자열 s의 길이 : 50 이하의 자연수

문자열 s는 알파벳으로만 이루어져 있습니다.

**입출력 예**



**풀이)**



따로 vsCode 로 옮겨서 테스트해보지 않아도 풀 정도로 간단한 문제였습니다만 풀이방식이 많이 지저분 했습니다.

위와같은 경우엔 1차원적으로 풀이하듯이 되었는데요

문자열은 길이를 추출할 수 있으니 넘겨받은 s의 길이만큼 반복해서

charAt() 함수를 사용하여 조건에 부합하는지 체크하는 방식으로 진행하였습니다.

아쉬운점은 toUpperCase(s) 같은 함수를 썼으면 조건식에 대소문자 두번 입력하는 일이 줄어들었을 것입니다.

두번째로 p 와 y 를 비교하기위헤 a,b 변수를 만들어 각각 카운트를 증가시켰는데요

알고리즘문제는 p,y의 갯수가 같거나 하나도없으면 true 아니면 false 이기 때문에

변수를 하나만 생성해서 y 일때는 ++ , p 일때는 -- 로 결과값이 0이냐 1이냐로 비교할수도 있었을 것 입니다.

**다른 사람의 풀이)**



볼때마다 대단한것 같습니다. s 라는 문자열을 모두 대문자로 치환 후 split 함수를 사용하여

P를 모두 잘라내었고 그 P와 Y의 길이를 잽니다.

그러고나서 바로 비교를 통해 true,false 를 반환시키는 깔끔한 코드가 나왔습니다.

split() 함수는 특정 문자 기준으로 잘라서 배열로 담아줍니다.



위와같은 문자열을 제공 받았을때 어떻게 잘리는지 Log를 찍어보았습니다.





pPoooyY 라는 문자열에서 P 기준으로 split 하게되면

처음 p를 만나서 그 앞까지 잘라주게 되는데 값이없으니 빈값이 담기게 됩니다.

두번째 p를 만나서 또 그 앞까지 잘라주게 되는데 이미 첫번쨰 P는 잘렸기 때문에 또 빈값을 담습니다.

그렇게 "", "", "나머지값"; 이 담긴채 끝나게 됩니다. Y도 마찬가지 입니다.

내장함수의 순기능을 잘 사용한것 같습니다.



두번째로는 위 함수가 추천을 많이 받았습니다.

match 와 정규식표현을 사용하여 코드를 짠 것 같습니다.

/p/ => p라는 문자를

i => 대소문자 구분없이

g => 모두 찾아라

라고하는 정규표현식 플래그 입니다.

그렇게 각 p 와 y를 찾아서 위와 마찬가지로 길이를 비교해 참, 거짓을 리턴하네요

그런데 문제가 있다고 합니다.

s 의 값에 P 와 Y 둘다 없을땐 match 함수는 null을 반환 합니다.

null은 length 함수를 가지고 있지 않기 때문에 에러가 났을겁니다.





우선 match의 실행결과 입니다.



위의 결과대로 match는 조건에 부합되는 값들을 배열형태로 담아줍니다.

그것의 길이를 재고 비교하면 true false가 나오게 됩니다.

\*문제는 p 와 y가 둘 다 없을땐 null 을 뱉기떄문에 length 함수를 쓸 수 없고 에러를 뱉게 됩니다.

**문제 링크주소** <https://programmers.co.kr/learn/courses/30/lessons/12916>