

유형	구현	소요 시간	30분						
문제	<p>[문제 설명]</p> <p>문자열 계산식을 입력받는 계산기를 만들려고 합니다. 계산식은 덧셈(+), 뺄셈(-), 숫자로만 이루어져 있습니다.</p> <p>계산식 s가 주어질 때, 계산식의 계산 결과를 반환하는 함수, solution을 완성해주세요.</p> <p>예를 들어, s가 '-3+26-7'로 주어질 때, 계산식의 계산 결과는 16 (-3 + 26 -7)입니다.</p>								
	<p>[제한 사항]</p> <p>- 계산식은 덧셈(+), 뺄셈(-), 숫자로만 이루어져 있습니다.</p>								
	<p>[입력 형식]</p> <p>- s는 길이가 1 이상 10 이하의 문자열입니다.</p>								
	<p>[출력 형식]</p> <p>계산식의 계산 결과를 출력합니다.</p>								
매개변수 & 리턴타입	<table><tr><th colspan="2">매개변수</th><th>Return</th></tr><tr><td>string</td><td>s</td><td>number</td></tr></table>			매개변수		Return	string	s	number
매개변수		Return							
string	s	number							
초기 코드	<pre>/**  * @param s {string}  * @return {number}  */ function solution(s) {   var answer = 0;   return answer; }</pre>								
모범답안	<pre>function solution(s) {   var nums = []; // 숫자 저장 공간   var ops = []; // 문자열 저장 공간   save(s, nums, ops);    return calc(nums, ops); }  function save(s, nums, ops) {   var i = 0;   var tmpNum = '';   while (i &lt; s.length) {     var c = s.charAt(i);</pre>								

```

    if (isNumber(c)) {
        tmpNum += c;
    } else {
        if (i === 0) {
            nums.push(0)
            ops.push(c);
        } else {
            nums.push(Number(tmpNum));
            ops.push(c);
        }
        tmpNum = '';
    }

    i++;
}

// 마지막 남은 숫자 추가
nums.push(Number(tmpNum));
}

function calc(nums, ops) {
    if (nums.length === 1) return nums[0];

    var num1 = nums.shift();
    var num2 = nums.shift();

    var calcNum = (ops.shift() === '+') ? add(num1, num2) :
sub(num1, num2);
    nums.unshift(calcNum)
    return calc(nums, ops)
}

function add(num1, num2) {
    return num1 + num2;
}

function sub(num1, num2) {
    return num1 - num2;
}

function isNumber(c) {
    var charCode = c.charCodeAt(0);
    return 48 <= charCode && charCode <= 57;
}

```

테스트  
케이스

예시케이스(1개)

입력		출력
s		
'-3+26-7'		16

정확성 테스트케이스(5개)

입력		출력
s		

