문제) 나누어 떨어지는 숫자 배열

array의 각 element 중 divisor로 나누어 떨어지는 값을 오름차순으로 정렬한 배열을 반환하는 함수, solution을 작성해주세요.

divisor로 나누어 떨어지는 element가 하나도 없다면 배열에 -1을 담아 반환하세요.

제한사항

* arr은 자연수를 담은 배열입니다.
* 정수 i, j에 대해 i ≠ j 이면 arr[i] ≠ arr[j] 입니다.
* divisor는 자연수입니다.
* array는 길이 1 이상인 배열입니다.

입출력 예

|  |  |  |
| --- | --- | --- |
| arr | divisor | return |
| [5, 9, 7, 10] | 5 | [5, 10] |
| [2, 36, 1, 3] | 1 | [1, 2, 3, 36] |
| [3, 2, 6] | 10 | [-1] |

입출력 예 설명

입출력 예#1

arr의 원소 중 5로 나누어 떨어지는 원소는 5와 10입니다. 따라서 [5, 10]을 리턴합니다.

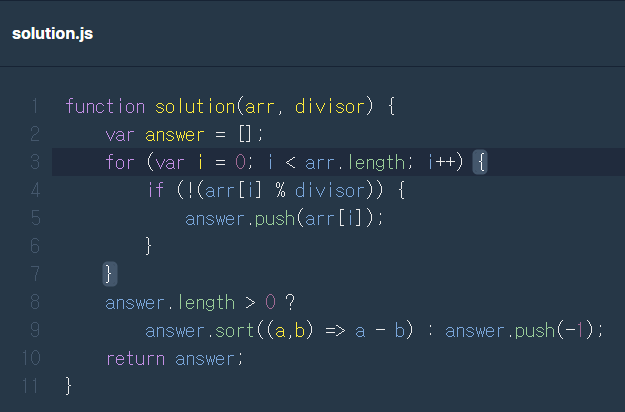
입출력 예#2

arr의 모든 원소는 1으로 나누어 떨어집니다. 원소를 오름차순으로 정렬해 [1, 2, 3, 36]을 리턴합니다.

입출력 예#3

3, 2, 6은 10으로 나누어 떨어지지 않습니다. 나누어 떨어지는 원소가 없으므로 [-1]을 리턴합니다.

solution)



arr 배열 안의 모든 원소에 대해 divisor로 나눈 나머지가 0이면 answer Array 안의 배열로 추가한다.

이때 if문의 조건에서 arr[i] % divisor가 0일 경우에 !0 은 true가 되어 if 문 내부 구문이 실행된다.(0 == false이므로)

만약 answer Array가 0보다 크다면 모든 원소를 오름차순으로 정렬.

answer Array 안에 원소가 없다면 -1을 추가한다.

다른 사람의 풀이)

function solution(arr, divisor) {

var answer = [ ];

arr.map((o) => {

o % divisor === 0 && answer.push(o);

});

return answer.length ? answer.sort((a, b) => a – b) : [-1];

}

map( ) 메서드는 배열 내의 모든 요소 각각에 대하여 주어진 함수를 호출한 결과를 모아 새로운 배열을 반환한다.

위 코드에서 arr.map 내부의 콜백함수에서는 각각의 element요소를 divisor로 나누었을 때 나머지가 0이 되는 경우에만 && 옆에 있는 구문이 실행되어 answer 배열에 그 element가 추가된다.

또한 answer.length 길이가 0인 경우에는 false이므로 [-1]이 함수의 결과값으로 반환되고 1 이상인 경우에는 true이므로 answer 배열의 원소들이 오름차순으로 정렬된 후 함수의 결과값으로 반환된다.