**문자열 내 마음대로 정렬하기**

**문제**

문자열로 구성된 리스트 strings와, 정수 n이 주어졌을 때, 각 문자열의 인덱스 n번째 글자를 기준으로 오름차순 정렬하려 합니다. 예를 들어 strings가 [sun, bed, car]이고 n이 1이면 각 단어의 인덱스 1의 문자 u, e, a로 strings를 정렬합니다.

**제한 조건**

strings는 길이 1 이상, 50이하인 배열입니다.  
strings의 원소는 소문자 알파벳으로 이루어져 있습니다.  
trings의 원소는 길이 1 이상, 100이하인 문자열입니다.  
모든 strings의 원소의 길이는 n보다 큽니다.  
인덱스 1의 문자가 같은 문자열이 여럿 일 경우, 사전순으로 앞선 문자열이 앞쪽에 위치합니다.

**입출력 예**

| **strings** | **n** | **return** |
| --- | --- | --- |
| [sun, bed, car] | 1 | [car, bed, sun] |
| [abce, abcd, cdx] | 2 | [abcd, abce, cdx] |

입출력 예 설명

입출력 예 1  
sun, bed, car의 1번째 인덱스 값은 각각 u, e, a 입니다. 이를 기준으로 strings를 정렬하면 [car, bed, sun] 입니다.

입출력 예 2  
abce와 abcd, cdx의 2번째 인덱스 값은 c, c, x입니다. 따라서 정렬 후에는 cdx가 가장 뒤에 위치합니다. abce와 abcd는 사전순으로 정렬하면 abcd가 우선하므로, 답은 [abcd, abce, cdx] 입니다.

**문제 링크주소 :** <https://programmers.co.kr/learn/courses/30/lessons/12915>

**풀이**

    function compare(*a*, *b*) {

        if (*a*.charAt(n) > *b*.charAt(n)) return 1;

        else if (*b*.charAt(n) > *a*.charAt(n)) return -1;

        else if (*a*.charAt(n) == *b*.charAt(n)) {

           return *a*.localeCompare(*b*);

        }

    }

    answer = strings.sort(compare);

    answer = strings.sort(compare);

sort함수에 compare함수를 인자로 넣어서, 정렬되도록 하였다.

        if (*a*.charAt(n) > *b*.charAt(n)) return 1;

        else if (*b*.charAt(n) > *a*.charAt(n)) return -1;

charAt함수로 n번째 오는 글자들을 비교하여, n번째글자를 기준으로 오름차순정렬되도록 설정하였다. (sort메소드는 return 값이 음수,양수,0인지에 따라서 순서를 정한다. )

retrun값이 1 즉, 양수를 return 하면 a의 n번째 값이 b의 n번째 값보다 크다는 의미이므로, 둘이 자리를 바꾸게 됩니다. ( a가 오른쪽으로 이동)

retrun값이 -1 즉, 음수를 return 하면 b의 n번째 값이 a의 n번째 값보다 크다는 의미이므로, 둘이 자리를 바꾸지 않습니다.

        else if (*a*.charAt(n) == *b*.charAt(n)) {

           return *a*.localeCompare(*b*);

        }

n번째 오는 글자가 같은 경우, localeCompare 이라는 메서드를 이용해서, 정렬하였다.

localeCompare() 메서드는 기준 문자열과 비교했을 때 비교 대상 문자열이 정렬상 전에 오는지, 후에 오는지 혹은 같은 순서에 배치되는지를 알려주는 숫자를 리턴한다. 즉 위와 마찬가지로 a가 b보다 크다면 양수값을 리턴하고, b가 a보다 크다면 음수값을 리턴합니다.