

## Coding Test

1. 2016년 1월 1일은 금요일입니다. 2016년 a월 b일은 무슨 요일일까요? 두 수 a ,b를 입력받아 2016년 a월 b일이 무슨 요일인지 리턴하는 함수, solution을 완성하세요. 요일의 이름은 일요일부터 토요일까지 각각 SUN,MON,TUE,WED,THU,FRI,SAT 입니다. 예를 들어 a=5, b=24라면 5월 24일은 화요일이므로 문자열 "TUE"를 반환하세요.

제한 조건

2016년은 윤년입니다.

2016년 a월 b일은 실제로 있는 날입니다.

(13월 26일이나 2월 45일같은 날짜는 주어지지 않습니다)

2. 단어 s의 가운데 글자를 반환하는 함수, solution을 만들어 보세요. 단어의 길이가 짝수라면 가운데 두글자를 반환하면 됩니다.

제한사항

s는 길이가 1 이상, 100이하인 스트링입니다.

입출력 예

"abcde" return "c"

"qwer" return "we"

3. 배열 arr가 주어집니다. 배열 arr의 각 원소는 숫자 0부터 9까지로 이루어져 있습니다. 이 때, 배열 arr에서 연속적으로 나타나는 숫자는 하나만 남기고 전부 제거하려고 합니다. 단, 제거된 후 남은 수들을 반환할 때는 배열 arr의 원소들의 순서를 유지해야 합니다. 예를 들면,

arr = [1, 1, 3, 3, 0, 1, 1] 이면 [1, 3, 0, 1] 을 return 합니다.

arr = [4, 4, 4, 3, 3] 이면 [4, 3] 을 return 합니다.

배열 arr에서 연속적으로 나타나는 숫자는 제거하고 남은 수들을 return 하는 solution 함수를 완성해 주세요.

제한사항

배열 arr의 크기 : 1,000,000 이하의 자연수

배열 arr의 원소의 크기 : 0보다 크거나 같고 9보다 작거나 같은 정수

입출력 예

arr [1, 1, 3, 3, 0, 1, 1]      answer [1, 3, 0, 1]

arr[4, 4, 4, 3, 3]              answer [4, 3]

4. 두 정수  $a$ ,  $b$ 가 주어졌을 때  $a$ 와  $b$  사이에 속한 모든 정수의 합을 리턴하는 함수, `solution`을 완성하세요.

예를 들어  $a = 3$ ,  $b = 5$ 인 경우,  $3 + 4 + 5 = 12$ 이므로 12를 리턴합니다.

제한 조건

$a$ 와  $b$ 가 같은 경우는 둘 중 아무 수나 리턴하세요.

$a$ 와  $b$ 는 -10,000,000 이상 10,000,000 이하인 정수입니다.

$a$ 와  $b$ 의 대소관계는 정해져있지 않습니다.

입출력 예

```
a 3   b 5   return 12
a 3   b 3   return  3
a 5   b 3   return 12
```

5. 문자열로 구성된 리스트 `strings`와, 정수  $n$ 이 주어졌을 때, 각 문자열의 인덱스  $n$ 번째 글자를 기준으로 오름차순 정렬하려 합니다. 예를 들어 `strings`가 `["sun", "bed", "car"]`이고  $n$ 이 1이면 각 단어의 인덱스 1의 문자 "u", "e", "a"로 `strings`를 정렬합니다.

제한 조건

`strings`는 길이 1 이상, 50이하인 배열입니다.

`strings`의 원소는 소문자 알파벳으로 이루어져 있습니다.

`strings`의 원소는 길이 1 이상, 100이하인 문자열입니다.

모든 `strings`의 원소의 길이는  $n$ 보다 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 1

"sun", "bed", "car"의 1번째 인덱스 값은 각각 "u", "e", "a" 입니다. 이를 기준으로 `strings`를 정렬하면 `["car", "bed", "sun"]` 입니다.

입출력 예 2

"abce"와 "abcd", "cdx"의 2번째 인덱스 값은 "c", "c", "x"입니다. 따라서 정렬 후에는 "cdx"가 가장 뒤에 위치합니다. "abce"와 "abcd"는 사전순으로 정렬하면 "abcd"가 우선하므로, 답은 `["abcd", "abce", "cdx"]` 입니다.