

02-2 문자열 처리 응용

1. 숫자 문자열과 영단어

<문제 설명>

네오와 프로도가 숫자놀이를 하고 있습니다. 네오가 프로도에게 숫자를 건넬 때 일부 자릿수를 영단어로 바꾼 카드를 건네주면 프로도는 원래 숫자를 찾는 게임입니다.

- 1478 → "one4seveneight"
- 234567 → "23four5six7"
- 10203 → "1zerotwozero3"

<제한 조건>

- $1 \leq s$ 의 길이 ≤ 50
- s 가 "zero" 또는 "0"으로 시작하는 경우는 주어지지 않습니다.
- return 값이 1 이상 2,000,000,000 이하의 정수가 되는 올바른 입력만 s 로 주어집니다.

숫자	영단어
0	zero
1	one
2	two
3	three
4	four
5	five
6	six
7	seven
8	eight
9	nine

<입출력 예>

s	return
"one4seveneight"	1478
"23four5six7"	234567
"2three45sixseven"	234567
"123"	123

입출력 예#3

"three"는 3, "six"는 6, "seven"은 7에 대응되기 때문에 정답은 입출력 예 #2와 같은 234567이 됩니다.

입출력 예 #2와 #3과 같이 같은 정답을 가리키는 문자열이 여러 가지가 나올 수 있습니다.

입출력 예#4

s 에는 영단어로 바뀐 부분이 없습니다.

02-2 문자열 처리 응용

2. 문자열 내림차순으로 배치하기

<문제 설명>

문자열 `s`에 나타나는 문자를 큰것부터 작은 순으로 정렬해 새로운 문자열을 리턴하는 함수, `solution`을 완성해주세요.

`s`는 영문 대소문자로만 구성되어 있으며, 대문자는 소문자보다 작은 것으로 간주합니다.

<제한 조건>

- `str`은 길이 1 이상인 문자열입니다.

<입출력 예>

s	return
"Zbcdefg"	"gfedcbZ"

02-2 문자열 처리 응용

3. 문자열 내 마음대로 정렬하기

<문제 설명>

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", "a"로 strings를 정렬합니다.

<제한 조건>

- strings는 길이 1 이상, 50이하인 배열입니다.
- strings의 원소는 소문자 알파벳으로 이루어져 있습니다.
- strings의 원소는 길이 1 이상, 100이하인 문자열입니다.
- 모든 strings의 원소의 길이는 n보다 큼니다.
- 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 strings를 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.

02-2 문자열 처리 응용

4. 이상한 문자 만들기

<문제 설명>

문자열 `s`는 한 개 이상의 단어로 구성되어 있습니다. 각 단어는 하나 이상의 공백문자로 구분되어 있습니다. 각 단어의 짝수번째 알파벳은 대문자로, 홀수번째 알파벳은 소문자로 바꾼 문자열을 리턴하는 함수, `solution`을 완성하세요.

<제한 조건>

- 문자열 전체의 짝/홀수 인덱스가 아니라, 단어(공백을 기준)별로 짝/홀수 인덱스를 판단해야 합니다.
- 첫 번째 글자는 0번째 인덱스로 보아 짝수번째 알파벳으로 처리해야 합니다.

<입출력 예>

s	return
"try hello world"	"TrY HeLlO WoRlD"

입출력 예

"try hello world"는 세 단어 "try", "hello", "world"로 구성되어 있습니다. 각 단어의 짝수번째 문자를 대문자로, 홀수번째 문자를 소문자로 바꾸면 "TrY", "HeLlO", "WoRlD"입니다. 따라서 "TrY HeLlO WoRlD" 를 리턴합니다.

02-2 문자열 처리 응용

5. 시저 암호

<문제 설명>

어떤 문장의 각 알파벳을 일정한 거리만큼 밀어서 다른 알파벳으로 바꾸는 암호화 방식을 시저 암호라고 합니다. 예를 들어 "AB"는 1만큼 밀면 "BC"가 되고, 3만큼 밀면 "DE"가 됩니다. "z"는 1만큼 밀면 "a"가 됩니다. 문자열 s와 거리 n을 입력받아 s를 n만큼 민 암호문을 만드는 함수, solution을 완성해 보세요.

<제한 조건>

- 공백은 아무리 밀어도 공백입니다.
- s는 알파벳 소문자, 대문자, 공백으로만 이루어져 있습니다.
- s의 길이는 8000이하입니다.
- n은 1 이상, 25이하인 자연수입니다.

<입출력 예>

s	n	result
"AB"	1	"BC"
"z"	1	"a"
"a B z"	4	"e F d"