

TEST 8

1.

카카오에 입사한 신입 개발자 남영이는 "카카오계정개발팀"에 배치되어, 카카오 서비스에 가입하는 유저들의 아이디를 생성하는 업무를 담당하게 되었습니다. 남영이에게 주어진 첫 업무는 새로 가입하는 유저들이 카카오 아이디 규칙에 맞지 않는 아이디를 입력했을 때, 입력된 아이디와 유사하면서 규칙에 맞는 아이디를 추천해주는 프로그램을 개발하는 것입니다.

다음은 카카오 아이디의 규칙입니다.

- > 아이디의 길이는 3자 이상 15자 이하여야 합니다.
- > 아이디는 알파벳 소문자, 숫자, 빼기(-), 밑줄(_), 마침표(.) 문자만 사용할 수 있습니다.
- > 단, 마침표(.)는 처음과 끝에 사용할 수 없으며 또한 연속으로 사용할 수 없습니다.

남영이는 다음과 같이 7단계의 순차적인 처리 과정을 통해 신규 유저가 입력한 아이디가 카카오 아이디 규칙에 맞는 지 검사하고 규칙에 맞지 않은 경우 규칙에 맞는 새로운 아이디를 추천해 주려고 합니다.

신규 유저가 입력한 아이디가 `new_id` 라고 한다면,

1단계 : `new_id`의 모든 대문자를 대응되는 소문자로 치환합니다.

2단계 : `new_id`에서 알파벳 소문자, 숫자, 빼기(-), 밑줄(_), 마침표(.)를 제외한 모든 문자를 제거합니다.

3단계 : `new_id`에서 마침표(.)가 2번 이상 연속된 부분을 하나의 마침표(.)로 치환합니다.

4단계 : `new_id`에서 마침표(.)가 처음이나 끝에 위치한다면 제거합니다.

5단계 : `new_id`가 빈 문자열이라면, `new_id`에 "a"를 대입합니다.

6단계 : `new_id`의 길이가 16자 이상이면, `new_id`의 첫 15개의 문자를 제외한 나머지 문자들을 모두 제거합니다. 만약 제거 후 마침표(.)가 `new_id`의 끝에 위치한다면 끝에 위치한 마침표(.) 문자를 제거합니다.

7단계 : `new_id`의 길이가 2자 이하라면, `new_id`의 마지막 문자를 `new_id`의 길이가 3이 될 때까지 반복해서 끝에 붙입니다.

TEST 8

예를 들어, new_id 값이 "...!@BaT#*..y.abcdefghijklm" 라면, 위 7단계를 거치고 나면 new_id 는 아래와 같이 변경됩니다.

1단계 대문자 'B'와 'T'가 소문자 'b'와 't'로 바뀌었습니다.

"...!@BaT#*..y.abcdefghijklm" → "...!@bat#*..y.abcdefghijklm"

2단계 '!', '@', '#', '*' 문자가 제거되었습니다.

"...!@bat#*..y.abcdefghijklm" → "...bat..y.abcdefghijklm"

3단계 '...'와 '..' 가 '.'로 바뀌었습니다.

"...bat..y.abcdefghijklm" → ".bat.y.abcdefghijklm"

4단계 아이디의 처음에 위치한 '.'가 제거되었습니다.

".bat.y.abcdefghijklm" → "bat.y.abcdefghijklm"

5단계 아이디가 빈 문자열이 아니므로 변화가 없습니다.

"bat.y.abcdefghijklm" → "bat.y.abcdefghijklm"

6단계 아이디의 길이가 16자 이상이므로, 처음 15자를 제외한 나머지 문자들이 제거되었습니다.

"bat.y.abcdefghijklm" → "bat.y.abcdefghi"

7단계 아이디의 길이가 2자 이하가 아니므로 변화가 없습니다.

"bat.y.abcdefghi" → "bat.y.abcdefghi"

TEST 8

제한사항

- ▶ new_id는 길이 1 이상 1,000 이하인 문자열입니다.
- ▶ new_id는 알파벳 대문자, 알파벳 소문자, 숫자, 특수문자로 구성되어 있습니다.
- ▶ new_id에 나타날 수 있는 특수문자는 `~!@#$%^&*()=+[{]}:?,<>/` 로 한정됩니다.

입출력 예

new_id	result
"...!@BaT#*..y.abcdefghijklm"	"bat.y.abcdefghi"
"z-+.^.."	"z--"
"=.="	"aaa"
"123_.def"	"123_.def"
"abcdefghijklmn.p"	"abcdefghijklmn"

#2>

1단계 변화 없습니다.

2단계 `"z-+.^.."` → `"z-.."`

3단계 `"z-.."` → `"z-."`

4단계 `"z-."` → `"z-"`

5단계 변화 없습니다.

6단계 변화 없습니다.

7단계 `"z-"` → `"z--"`

TEST 8

2.

세 차례의 코딩 테스트와 두 차례의 면접이라는 기나긴 블라인드 공채를 무사히 통과해 카카오에 입사한 응경이는 파일 저장소 서버 관리를 맡게 되었습니다.

저장소 서버에는 프로그램의 과거 버전을 모두 담고 있어, 이름 순으로 정렬된 파일 목록은 보기가 불편했습니다. 파일을 이름 순으로 정렬하면 나중에 만들어진 ver-10.zip 이 ver-9.zip 보다 먼저 표시되기 때문입니다.

버전 번호 외에도 숫자가 포함된 파일 목록은 여러 면에서 관리하기 불편했습니다. 예컨대 파일 목록이 [img12.png, img10.png, img2.png, img1.png] 일 경우,

일반적인 정렬은 [img1.png, img10.png, img12.png, img2.png] 순이 되지만,

숫자 정렬인 [img1.png, img2.png, img10.png, img12.png] 순이 훨씬 자연스럽습니다..

응경이는 단순한 문자 코드 순이 아닌, 파일명에 포함된 숫자를 반영한 정렬 기능을 저장소 관리 프로그램에 구현하기로 했습니다.

소스 파일 저장소에 저장된 파일명은 100 글자 이내로, 영문 대소문자, 숫자, 공백(" "), 마침표("."), 빼기 부호("-")만으로 이루어져 있다. 파일명은 영문자로 시작하며, 숫자를 하나 이상 포함하고 있습니다.

TEST 8

파일명은 크게 HEAD, NUMBER, TAIL 의 세 부분으로 구성됩니다.

- > HEAD는 숫자가 아닌 문자로 이루어져 있으며, 최소한 한 글자 이상입니다.
- > NUMBER는 한 글자에서 최대 다섯 글자 사이의 연속된 숫자로 이루어져 있으며, 앞쪽에 0이 올 수 있습니다.
- > TAIL은 그 나머지 부분으로, 여기에는 숫자가 다시 나타날 수도 있으며, 아무 글자도 없을 수 있습니다.

파일명	HEAD	NUMBER	TAIL
foo9.txt	foo	9	.txt
foo010bar020.zip	foo	010	bar020.zip
F-15	F-	15	(빈 문자열)

파일명을 세 부분으로 나눈 후, 다음 기준에 따라 파일명을 정렬합니다.

- > 파일명은 우선 HEAD 부분을 기준으로 사전 순으로 정렬합니다. 이때, 문자열 비교 시 대소문자 구분을 하지 않습니다. EUN 과 EuN, eun 은 정렬 시에 같은 수서로 취급됩니다.
- > 파일명의 HEAD 부분이 대소문자 차이 외에는 같을 경우, NUMBER의 숫자 순으로 정렬합니다. 9 < 10 < 0011 < 012 < 13 < 014 순으로 정렬됩니다. 숫자 앞의 0은 무시되며, 012와 12는 정렬 시에 같은 값으로 처리됩니다.
- > 두 파일의 HEAD 부분과, NUMBER 의 숫자도 같을 경우, 원래 입력에 주어진 순서를 유지합니다. EUN01.zip 과 eun1.png 가 입력으로 들어오면, 정렬 후에도 입력 시 주어진 두 파일의 순서가 바뀌어서는 안됩니다.

은경이를 도와 파일명 정렬 프로그램을 완성해주세요.

TEST 8

제한사항

▶ 입력으로 배열 `files`가 주어집니다.

> `files` 는 1000개 이하의 파일명을 포함하는 문자열 배열입니다.

> 각 파일명은 100 글자 이하 길이로, 영문 대소문자, 숫자, 공백, 마침표, 빼기 부호만으로 이루어져 있습니다. 파일명은 영문자로 시작하며, 숫자를 하나 이상 포함하고 있습니다.

> 중복된 파일명은 없으나, 대소문자나 숫자 앞부분의 0 차이가 있는 경우는 함께 주어질 수 있습니다. (`eun1.txt`, `EUN1.txt`, `eun001.txt`, `eun1.TXT` 는 함께 주어질 수 있습니다.)

files	result
[<code>img12.png</code> , <code>img10.png</code> , <code>img2.png</code> , <code>img1.png</code>]	[<code>img1.png</code> , <code>img2.png</code> , <code>img10.png</code> , <code>img12.png</code>]
[<code>"F-5 Freedom Fighter"</code> , <code>"B-50 Superfortress"</code> , <code>"A-10 Thunderbolt II"</code> , <code>"F-14 Tomcat"</code>]	[<code>"A-10 Thunderbolt II"</code> , <code>"B-50 Superfortress"</code> , <code>"F-5 Freedom Fighter"</code> , <code>"F-14 Tomcat"</code>]