

## 문제 설명

셀수있는 수량의 순서있는 열거 또는 어떤 순서를 따르는 요소들의 모음을 튜플(tuple)이라고 합니다. n개의 요소를 가진 튜플을 n-튜플(n-tuple)이라고 하며, 다음과 같이 표현할 수 있습니다.

- $(a_1, a_2, a_3, \dots, a_n)$

튜플은 다음과 같은 성질을 가지고 있습니다.

1. 중복된 원소가 있을 수 있습니다. ex :  $(2, 3, 1, 2)$
2. 원소에 정해진 순서가 있으며, 원소의 순서가 다르면 서로 다른 튜플입니다. ex :  $(1, 2, 3) \neq (1, 3, 2)$
3. 튜플의 원소 개수는 유한합니다.

원소의 개수가 n개이고, **중복되는 원소가 없는** 튜플  $(a_1, a_2, a_3, \dots, a_n)$ 이 주어질 때(단,  $a_1, a_2, \dots, a_n$ 은 자연수), 이는 다음과 같이 집합 기호 '{', '}'를 이용해 표현할 수 있습니다.

- $\{a_1\}, \{a_1, a_2\}, \{a_1, a_2, a_3\}, \{a_1, a_2, a_3, a_4\}, \dots \{a_1, a_2, a_3, a_4, \dots, a_n\}$

예를 들어 튜플이  $(2, 1, 3, 4)$ 인 경우 이는

- $\{2\}, \{2, 1\}, \{2, 1, 3\}, \{2, 1, 3, 4\}$

와 같이 표현할 수 있습니다. 이때, 집합은 원소의 순서가 바뀌어도 상관없으므로

- $\{2\}, \{2, 1\}, \{2, 1, 3\}, \{2, 1, 3, 4\}$
- $\{2, 1, 3, 4\}, \{2\}, \{2, 1, 3\}, \{2, 1\}$
- $\{1, 2, 3\}, \{2, 1\}, \{1, 2, 4, 3\}, \{2\}$

는 모두 같은 튜플  $(2, 1, 3, 4)$ 를 나타냅니다.

특정 튜플을 표현하는 집합이 담긴 문자열 s가 매개변수로 주어질 때, s가 표현하는 튜플을 배열에 담아 return 하도록 solution 함수를 완성해주세요.

## [제한사항]

- s의 길이는 5 이상 1,000,000 이하입니다.
- s는 숫자와 '{', '}', ',' 로만 이루어져 있습니다.
- 숫자가 0으로 시작하는 경우는 없습니다.
- s는 항상 중복되는 원소가 없는 튜플을 올바르게 표현하고 있습니다.
- s가 표현하는 튜플의 원소는 1 이상 100,000 이하인 자연수입니다.
- return 하는 배열의 길이가 1 이상 500 이하인 경우만 입력으로 주어집니다.

## [입출력 예]

s	result
"{{2},{2,1},{2,1,3},{2,1,3,4}}"	[2, 1, 3, 4]
"{{1,2,3},{2,1},{1,2,4,3},{2}}"	[2, 1, 3, 4]
"{{20,111},{111}}"	[111, 20]
"{{123}}"	[123]

s	result
"{{4,2,3},{3},{2,3,4,1},{2,3}}"	[3, 2, 4, 1]

## 입출력 예에 대한 설명

## 입출력 예 #1

문제 예시와 같습니다.

## 입출력 예 #2

문제 예시와 같습니다.

## 입출력 예 #3

(111, 20)을 집합 기호를 이용해 표현하면 {{111}, {111,20}}이 되며, 이는 {{20,111},{111}}과 같습니다.

## 입출력 예 #4

(123)을 집합 기호를 이용해 표현하면 {{123}} 입니다.

## 입출력 예 #5

(3, 2, 4, 1)을 집합 기호를 이용해 표현하면 {{3},{3,2},{3,2,4},{3,2,4,1}}이 되며, 이는 {{4,2,3},{3},{2,3,4,1},{2,3}}과 같습니다.

## solution.cpp

```

1  #include <string>
2  #include <vector>
3
4  using namespace std;
5
6  vector <string> sort(vector <string> s) {
7      for (int i = 0; i < s.size(); i++) {
8          for (int j = 0; j < s.size()-i-1; j++) {
9              if (s[j].size() > s[j+1].size()){
10                 string tmp = s[j];
11                 s[j] = s[j+1];
12                 s[j+1] = tmp;
13             }
14         }
15     }
16     return s;
17 }
18
19 vector<int> solution(string s) {
20     vector<int> answer;
21
22     vector <string> ss;
23     int n = 1, tmp = 0;
24     while(1) {
25         int a = s.find("{", n);
26         int b = s.find("}", n+1);
27         n = b;
28         ss.push_back(s.substr(a + 1, b - a));
29         if (s.at(n + 1) == '}') break;
30     }
31     ss = sort(ss);

```

```
32
33     for (int i = 0; i < ss[0].size(); i++) { //첫번째 값
34         if ('0' <= ss[0][i] && ss[0][i] <= '9') {
35             tmp = tmp * 10 + (int)ss[0][i] - '0';
36         }
37         else {
38             answer.push_back(tmp);
39             tmp = 0;
40         }
41     }
42     for (int i = 1; i < ss.size(); i++) { //두번째 이후 answer에 저장된 것과 중복되지 않는 수 저장
```

---

실행 결과

---

테스트 1
입력값 > "{2},{2,1},{2,1,3},{2,1,3,4}}"
기댓값 > [2, 1, 3, 4]
실행 결과 > 테스트를 통과하였습니다.
테스트 2
입력값 > "{1,2,3},{2,1},{1,2,4,3},{2}}"
기댓값 > [2, 1, 3, 4]
실행 결과 > 테스트를 통과하였습니다.
테스트 3
입력값 > "{20,111},{111}}"
기댓값 > [111, 20]
실행 결과 > 테스트를 통과하였습니다.
테스트 4
입력값 > "{{123}}"
기댓값 > [123]
실행 결과 > 테스트를 통과하였습니다.
테스트 5
입력값 > "{4,2,3},{3},{2,3,4,1},{2,3}}"
기댓값 > [3, 2, 4, 1]
실행 결과 > 테스트를 통과하였습니다.

테스트 결과 (~ˇ~)~

5개 중 5개 성공

샘플 테스트 케이스를 통과했다는 의미로, 작성한 코드가 문제의 정답은 아닐 수 있습니다.  
(샘플 테스트 케이스는 [테스트 케이스 추가하기] 버튼을 통해 확인하실 수 있습니다.)

- 초기화
- 실행
- 코드 채점하고 제출