

2022 LG CNS Code Monster 예선 for 한동곤

[시험 규정](#) [Python3 레퍼런스](#) [컴파일 옵션](#) [테스트 종료](#)

>



1



2



3



4

<

1. 프로그래밍

>

🕒 종료까지

00:13:20

🔍 프로그래밍

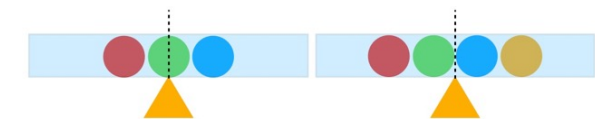
다시 풀 문제 🚩

문제 설명

동일한 모양의 구슬 n 개, 구슬들을 일렬로 배열할 수 있는 무한히 긴 유리관 시소가 있습니다. 당신은 구슬과 유리관시소를 사용해 **가장 아름다운 구슬 장식**을 만들려고 합니다. **구슬마다 무게가 다를 수 있습니다.**

구슬 장식을 만드는 방법은 다음과 같습니다.

- 원하는 구슬들을 유리관 안에 넣습니다.



- 구슬은 위 그림처럼 시소 받침대 바로 위에 구슬의 중앙이 오거나, 시소 받침대를 기준으로 좌우에 오도록 배치합니다.
- 시소의 받침대를 기준으로 왼쪽에 위치한 구슬들의 무게의 합과 오른쪽에 위치한 구슬들의 무게의 합이 같아야 합니다. 중앙에 놓인 구슬이 있다면 좌우 무게를 계산할 때만 제외합니다.

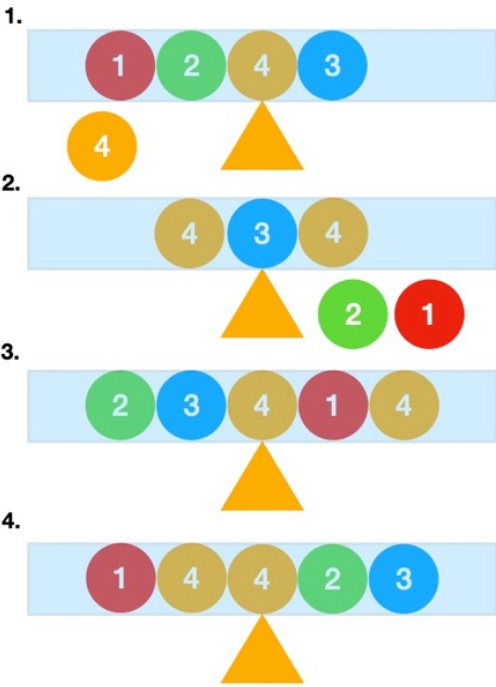
구슬 장식이 갖는 아름다움의 정도를 비교하는 방법은 다음과 같습니다.

먼저 1번 방법을 사용해서 비교하고, 우위를 가릴 수 없다면 번호 순서대로 다음 방법을 사용하여 비교합니다.

- 시소 받침대를 기준으로 좌우 구슬 수의 차이가 적을수록 더 아름답습니다.

- 2. 더 많은 구슬을 사용한 장식이 더 아름답습니다.
- 3. 구슬의 무게 합이 더 무거운 장식이 더 아름답습니다.
- 4. 장식에 사용된 구슬의 무게들을 왼쪽부터 순서대로 담아 만든 배열이 사전 순으로 더 빠른 장식이 더 아름답습니다. (첫 번째 원소부터 순서대로 비교해서 작은 원소가 먼저 나오거나 비교할 원소가 먼저 바닥난 배열이 사전 순으로 더 빠른 배열입니다.)

다음은 구슬 장식을 만드는 예시입니다.



위 그림은 [1, 2, 3, 4, 4] 무게를 가진 구슬 다섯 개로 만들 수 있는 장식 중 일부입니다. 그림에 나타낸 네 가지 장식들은 4, 3, 2, 1 순으로 더 아름답습니다.

- 1번 장식과 2번 장식을 비교하면 1번 조건에 따라 좌우 구슬의 개수 차이가 적은 2번 장식이 더 아름답습니다.
- 2번 장식과 3번 장식을 비교하면 1번 조건은 같지만 2번 조건에 따라 더 많은 구슬을 사용한 3번 장식이 더 아름답습니다.
- 3번 장식과 4번 장식을 비교하면 1, 2, 3번 조건은 같지만 4번 조건에 따라 구슬의 무게로 만든 배열이 사전 순으로 빠른 4번 장식이 더 아름답습니다.

구슬 n 개의 무게를 의미하는 1차원 정수 배열 marbles 가 매개변수로 주어집니다. 이때 주어진 조건에 따라 만들 수 있는 가장 아름다운 장식에 사용된 구슬의 순서를 1차원 정수 배열에 담아 return 하도록 solution 함수를 완성해주세요.

제한사항

- 1 ≤ marbles 의 길이 = 구슬의 개수 n ≤ 9
 - 1 ≤ marbles 의 원소 ≤ 10

- `marbles[i]` 는 `i + 1` 번 구슬의 무게를 나타냅니다.

입출력 예

marbles	result
[1, 2, 3, 4, 4]	[1, 4, 4, 2, 3]
[5, 5, 1, 4]	[5, 4, 5]
[3, 9, 7, 5]	[3, 9, 5, 7]
[7, 3, 1]	[7]

입출력 예 설명

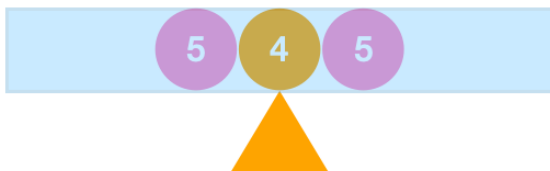
입출력 예 #1

문제 예시와 같습니다.

가장 아름다운 장식은 구슬을 `[1, 4, 4, 2, 3]` 순서로 배치한 모양입니다.

따라서 `[1, 4, 4, 2, 3]` 을 return 합니다.

입출력 예 #2



`[5, 5, 1, 4]` 무게의 구슬들로 `[5, 5, 1, 4]` , `[5, 1, 4]` , `[5, 1, 5]` , `[5, 4, 5]` 등 여러 형태의 장식을 만들 수 있습니다. 그중 가장 아름다운 장식은 위 그림처럼 구슬을 `[5, 4, 5]` 순서로 배치한 모양입니다.

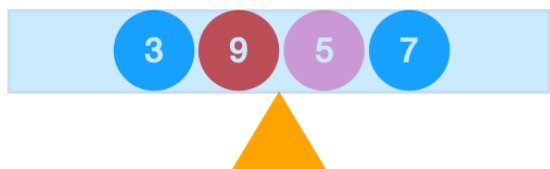
`[5, 5, 1, 4]` , `[5, 1, 4]` 순서는 1번 조건에 따라 좌우 구슬의 개수 차이가 큼니다.

`[5, 5]` , `[1]` 등은 2번 조건에 따라 사용한 구슬의 개수가 더 적습니다.

`[5, 1, 5]` 순서는 1, 2번 조건은 같지만 3번 조건에 따라 구슬의 무게 합이 작습니다.

따라서 `[5, 4, 5]` 를 return 합니다.

입출력 예 #3



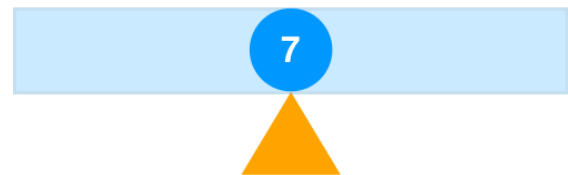
`[3, 9, 7, 5]` 무게의 구슬들로 `[3, 9, 5, 7]` , `[3, 9, 7, 5]` , `[5, 7, 3, 9]` 등 여러 형태의 장식을 만들 수 있습니다. 그중 가장 아름다운 장식은 위 그림처럼 구슬을

[3, 9, 5, 7] 순서로 배치한 모양입니다.

[3, 9, 5, 7], [3, 9, 7, 5], [5, 7, 3, 9] 등 구슬 4개를 모두 사용해 시소의 균형을 맞출 수 있습니다. 이 장식들은 1, 2, 3번 조건이 모두 같습니다. 더 아름다운 장식은 4번 조건에 따라 구슬의 무게들을 왼쪽부터 순서대로 담아 만든 배열이 사전 순으로 더 빠른 장식입니다.

따라서 [3, 9, 5, 7] 을 return 합니다.

입출력 예 #4



[7, 3, 1] 무게의 구슬들 중 2개 이상을 사용해 시소의 균형을 맞출 수 없습니다. 구슬을 하나만 사용한 [1], [3], [7] 순서의 장식 중 3번 조건에 따라 가장 무거운 [7] 이 가장 아름답습니다.

따라서 [7] 을 return 합니다.

제한시간 안내

- 각 테스트 케이스의 제한시간은 10초입니다.

Python3 ✓ 2시간 전 저장됨

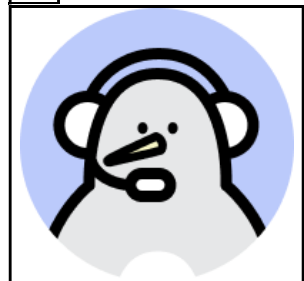
```
from itertools import permutations, combinations
from functools import cmp_to_key

def solution(marbles):
    answer = []
    for count in range(1, len(marbles)+1):
        for permutation in permutations(marbles, count):
            left, right = 0, len(permutation) - 1
            leftSum, rightSum = 0, 0
            while left <= right:
                if left == right and leftSum == rightSum:
                    break
                if leftSum <= rightSum:
                    leftSum += permutation[left]
                    left += 1
                else:
                    rightSum += permutation[right]
                    right -= 1
            if leftSum == rightSum:
```

실행 결과
실행 결과가 여기에 표시됩니다.

테스트 케이스 추가하기 내 답안 코드 초기화 코드 실행 코드 제출하기





:-(
화면 크기가 너무 작습니다.

브라우저 창을 최대로 키워주시고, 이미 최대라면 글꼴 크기를 줄여주세요. 시험 응시에 필요한 최소 화면 너비는 가로 992px 입니다.