华为OD机试 - 跳房子I（Java & JS & Python）

题目描述

跳房子，也叫跳飞机，是一种世界性的儿童游戏。

游戏参与者需要分多个回合按顺序跳到第1格直到房子的最后一格。

跳房子的过程中，可以向前跳，也可以向后跳。

假设房子的总格数是count，小红每回合可能连续跳的步教都放在数组steps中，请问数组中是否有一种步数的组合，可以让小红两个回合跳到量后一格？

如果有，请输出索引和最小的步数组合。

注意：

- 数组中的步数可以重复，但数组中的元素不能重复使用。
- 提供的数据保证存在满足题目要求的组合，且索引和最小的步数组合是唯一的。

输入描述

第一行输入为房子总格数count，它是int整数类型。
第二行输入为每回合可能连续跳的步数，它是int整数数组类型。

输出描述

返回索引和最小的满足要求的步数组合（顺序保持steps中原有顺序）

备注

- count ≤ 1000
- 0 ≤ steps.length ≤ 5000
- -100000000 ≤ steps ≤ 100000000

用例

输入	[1,4,5,2,2] 7
输出	[5, 2]
说明	无

输入	[-1,2,4,9,6] 8
输出	[-1, 9]
说明	此样例有多种组合满足两回合跳到最后，譬如：[-1,9]，[2,6]，其中[-1,9]的索引和为0+3=3，[2,6]的索和为1+4=5，所以索引和最小的步数组合[-1,9]

题目解析

本题其实就是两数之和：LeetCode - 1 两数之和_伏城之外的博客-CSDN博客的变种题。

即有一个数组steps，要在其中找到一个二元组，让其和等于count。

和leetcode两数之和的区别在于，本题最终解二元组可能有多个解，我们需要在这多个解中找到索引和最小的作为最终解，即我们不仅要求二元组元素之和，还要求二元组索引之和。

本题解析可以参考上面链接博客。

```java
import java.util.Arrays;
import java.util.HashMap;
import java.util.Scanner;

public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    String tmp = sc.nextLine();
    int[] steps =
        Arrays.stream(tmp.substring(1, tmp.length() - 1).split(","))
            .mapToInt(Integer::parseInt)
            .toArray();

    int count = Integer.parseInt(sc.nextLine());

    System.out.println(getResult(steps, count));
  }

  public static String getResult(int[] steps, int count) {
    HashMap<Integer, Integer> map = new HashMap<>();
```

```
22
23      int minIdxSum = Integer.MAX_VALUE;
24      String ans = "";
25
26      for (int idx1 = 0; idx1 < steps.length; idx1++) {
27        int step1 = steps[idx1];
28        int step2 = count - step1;
29
30        if (map.containsKey(step2)) {
31          int idx2 = map.get(step2);
32          int idxSum = idx1 + idx2;
33          if (idxSum < minIdxSum) {
34            minIdxSum = idxSum;
35            ans = idx1 < idx2 ? "[" + step1 + ", " + step2 + "]" : "[" + step2 + ", " +
    step1 + "]";
36          }
37
38        } else {
39          map.put(step1, idx1);
40        }
41      }
42
43      return ans;
44    }
45  }
```