

题目描述

一贫如洗的樵夫阿里巴巴在去砍柴的路上，无意中发现了强盗集团的藏宝地，藏宝地有编号从0~N的箱子，每个箱子上面贴有一个数字，箱子~~内~~可能有一个黄金宝箱。

黄金宝箱满足排在它之前的所有箱子数字和等于排在它之后的所有箱子数字之和；

第一个箱子左边部分的数字和定义为0；最后一个箱子右边部分的数字和定义为0.

请帮阿里巴巴找到黄金宝箱，输出第一个满足条件的黄金宝箱编号，如果不存在黄金宝箱，请返回-1。

输入描述

箱子上贴的数字列表，使用逗号分隔，例如1, -1, 0

宝箱的数量不小于1个，不超过10000

宝箱上贴的数值范围不低于-1000，不超过1000

输出描述

第一个黄金宝箱的编号

用例

输入	2,5,-1,8,6
输出	3
说明	下标3之前的数字和为： $2 + 5 + -1 = 6$ 下标3之后的数字和为：6 = 6

输入	8,9
输出	-1
说明	不存在符合要求的位置

输入	11
输出	0
说明	下标0之前的数字和为：0 下标0之后的数字和为：0

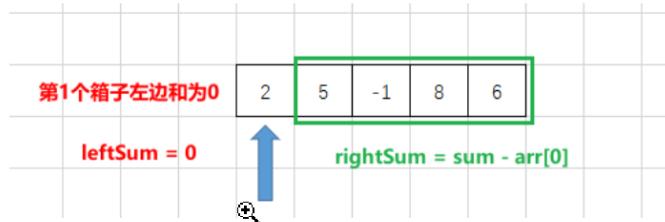
题目解析

本题可以利用单指针来解决。

我的解题思路如下：

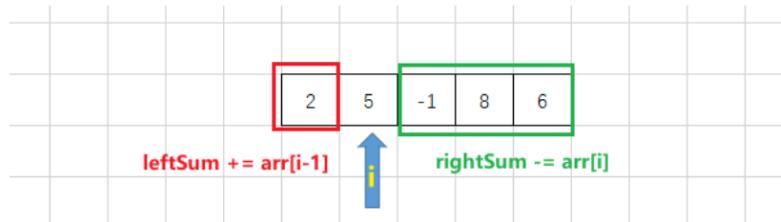
首先求出输入的数组arr的所有元素之和sum，然后用一个指针i从0开始遍历数组arr，假设指针i左边元素之和为leftSum，指针i右边元素之和为rightSum。

那么初始时，即*i = 0*时，*leftSum = 0*，*rightSum = sum - arr[0]*，如下图：



如果此时*leftSum == rightSum*，那么*i = 0*就是宝箱位置。

否则，继续*i++*，之后*leftSum += arr[i-1]*, *rightSum -= arr[i]*



如果*i*走到最后一个元素位置，还没有达成*leftSum == rightSum*，那么就返回-1。

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int[] arr =
9             Arrays.stream(sc.nextLine().split(",")).mapToInt(Integer::parseInt).toArray();
10
11     }
12
13     public static int getResult(int[] arr) {
14         int leftSum = 0;
15         int rightSum = Arrays.stream(arr).sum() - arr[0];
16
17         if (leftSum == rightSum) return 0;
```

```
18
19  for (int i = 1; i < arr.length; i++) {
20      leftSum += arr[i - 1];
21      rightSum -= arr[i];
22      if (leftSum == rightSum) return i;
23  }
24
25  return -1;
26}
27}
```