

阿里巴巴找黄金宝箱(I)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 $0 \sim N$ 的箱子，每个箱子上面贴有一个数字，箱子中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和；第一个箱子左边部分的数字和定义为 0；最后一个宝箱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

补充说明：

示例 1

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： $6 = 6$

示例 2

输入：

8,9

输出：

-1

说明：

不存在符合要求的位置

示例 3

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

```
import java.util.*;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
```

```
        // 注意 hasNext 和 hasNextLine 的区别
```

```
        while (in.hasNextLine()) { // 注意 while 处理多个 case
```

```
            int ans = -1;
```

```
            int[] arr = Arrays.stream(in.nextLine().split(","))
```

```

.mapToInt(Integer::parseInt).toArray();

int[] leftSum = new int[arr.length], rightSum = new int[arr.length];

int i;

for (i = 1; i < arr.length-1; i++){

    leftSum[i] = leftSum[i-1]+arr[i-1];

    rightSum[arr.length-1-i] =
rightSum[arr.length-i]+arr[arr.length-i];

}

if (arr.length > 1){

    leftSum[i] = leftSum[i-1]+arr[i-1];

    rightSum[arr.length-1-i] =
rightSum[arr.length-i]+arr[arr.length-i];

}

for (i = 0; i < arr.length; i++){

    if (leftSum[i] == rightSum[i]){

        ans = i;

        break;

    }

}

System.out.println(ans);

}

}

```

}