

题目描述：

一个荒岛上有若干人，岛上只有一条路通往岛屿两端的港口，大家需要逃往两端的港口才可逃生。假定每个人移动的速度一样，且只可选择向左或 向右逃生。若两个人相遇，则进行决斗，战斗力强的能够活下来，并损失掉与对方相同的战斗力；若战斗力相同，则两人同归于尽。

输入描述：

给定一非 0 整数数组，元素个数不超过 30000；正负表示逃生方向（正表示向右逃生，负表示向左逃生），绝对值表示战斗力，越左边的数字表示离左边港口越近，逃生方向相同的人永远不会发生决斗。

输出描述：

能够逃生的人总数,没有人逃生输出 0，输入异常时输出-1。

补充说明：

示例1

输入：5 10 8 -8 -5

输出：2

说明：第3个人和第4个人同归于尽，第2个人杀死第5个人并剩余5战斗力，第1个人没有遇到敌人。

```

1  import java.util.*;
2
3  public class Main {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          int[] arr = Arrays.stream(scanner.nextLine().split(" ")).mapToInt(
7              Integer::parseInt).toArray();
8          int liveMan = asteroidCollision(arr);
9          System.out.println(liveMan);
10     }
11
12     static int asteroidCollision(int[] arr) {
13         int n = arr.length;
14         int p = 0;
15         Deque<Integer> stack = new LinkedList<>();
16         while (p < n) {
17             if (stack.isEmpty() || stack.peek() < 0 || arr[p] > 0) {
18                 stack.push(arr[p]);
19             } else if (stack.peek() <= -arr[p]) {
20                 int cur = stack.pop();
21                 if (cur < -arr[p]) {
22                     int tem = -arr[p] - cur;
23                     while (!stack.isEmpty() && stack.peek() > 0 && tem >= stack.peek()) {
24                         int t = stack.pop();
25                         tem -= t;
26                     }
27                     if (!stack.isEmpty()) {
28                         if (stack.peek() < 0) {
29                             if (tem > 0) {
30                                 stack.push(-tem);
31                             }
32                         } else if (stack.peek() > tem) {
33                             stack.push(stack.pop() - tem);
34                         }
35                     } else {
36                         if (tem > 0) {
37                             stack.push(-tem);
38                         }
39                     }
40                 }
41             } else if (stack.peek() > -arr[p]) {
42                 int cur = stack.pop();
43                 stack.push(cur - -arr[p]);
44             }
45             p++;
46         }
47         return stack.size();
48     }
49 }

```