

## 一、编程题

### ACM：机器人仓库搬砖

题目描述：机器人搬砖，一共有N堆砖存放在N个不同的仓库中，第i堆砖中有bricks[i]块砖头，要求在8小时内搬完。机器人每小时能搬砖的数量取决于有多少能量格，机器人一个小时中只能在一个仓库中搬砖，机器人的能量格每小时补充一次且能量格只在这一个小时有效，为使得机器人损耗最小化尽量减小每次补充的能量格数。

为了保障在8小时内能完成搬砖任务，请计算每小时给机器人充能的最小能量格数。

备注：

- 1、无需考虑机器人补充能量格的耗时；
- 2、无需考虑机器人搬砖的耗时；
- 3、机器人每小时补充能量格只在这一个小时中有效；

输入描述：程序输入为“30 12 25 8 19”一个整数数组，数组中的每个数字代表第i堆砖的个数，每堆砖的个数不超过100。

输出描述：输出在8小时内完成搬砖任务，机器人每小时最少需要充多少个能量格；  
如果8个小时内无论如何都完成不了任务，则输出“-1”；

补充说明：

示例1

输入：30 12 25 8 19

输出：15

说明：

示例2

输入：10 12 25 8 19 8 6 4 17 19 20 30

输出：-1

说明：砖的堆数为12堆存放在12个仓库中，机器人一个小时内只能在一个仓库搬砖，不可能完成任务。

代码：

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

```
            String s = in.nextLine();
```

```
            String[] s1 = s.split(" ");
```

```
            int[] bricks = new int[s1.length];
```

```
            for (int i = 0; i < s1.length; i++) {
```

```
                bricks[i] = Integer.parseInt(s1[i]);
```

```
            }
```

```
            Arrays.sort(bricks);
```

```
            if (bricks.length > 8) {
```

```
                System.out.println(-1);
```

```
                continue;
```

```
            } else if (bricks.length == 8) {
```

```
                System.out.println(bricks[bricks.length - 1]);
```

```
                continue;
```

```
            }
```

```
// 还剩的空余次数
int times = 8 - bricks.length;
// int[] newBricks = new int[8];
List<Integer> list = new ArrayList<>();
for (int i = 0; i < times && i < bricks.length; i++) {
    int num = bricks[bricks.length - 1 - i] / 2;
    list.add(num);
    list.add(bricks[bricks.length - 1 - i] - num);
}
for (int i = 0; i < bricks.length - times; i++) {
    list.add(bricks[i]);
}
Integer max = Collections.max(list);
System.out.println(max);

    }
}
}
```