

java-小明和朋友玩跳格子游戏

题目描述：

小明和朋友玩跳格子游戏，有  $n$  个连续格子，每个格子有不同的分数，小朋友可以选择

从任意格子起跳，但是不能跳连续的格子，也不能回头跳；

给定一个代表每个格子得分的非负整数数组，计算能够得到的最高分数。

输入描述：

给定一个数例，如：

1 2 3 1

输出描述：

输出能够得到的最高分，如：

4

补充说明：

$1 \leq \text{nums.length} \leq 100$

$0 \leq \text{nums}[i] \leq 1000$

示例 1

输入：

1 2 3 1

输出：

4

说明：

选择跳第一个格子和第三个格子

示例 2

输入：

2 7 9 3 1

输出：

12

说明：

$2+9+1=12$

```
import java.util.Scanner;
```

```
import java.util.ArrayList;
```

```
import java.util.LinkedList;
```

```
import java.util.Arrays;
```

```
import java.util.HashMap;
```

```
// 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);
```

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

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

```
        System.out.println(function(scanInt));
```

```
    }
```

```
    public static int function(int[] scanInt) {
```

```
        int soud = scanInt.length;
```

```
        int[] moo = new int[soud];
```

```
        if(soud>=1) moo[0] = scanInt[0];
```

```
        if(soud>=2) moo[1] = Math.max(scanInt[0],scanInt[1]);
```

```
        for(int i=2;i<soud;i++) {
```

```
moo[i] = Math.max(moo[i-1],moo[i-2]+scanInt[i]);
```

```
}
```

```
return moo[soud-1];
```

```
}
```

```
}
```