

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.*;
public class Main {
    private static int [] arr = null;
    private static int [] score = null;
    private static HashMap<Integer, Integer> map = new HashMap();
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        while (in.hasNext()) { // 注意，如果输入是多个测试用例，请通过 while 循环处理多个测试用例
            String s = in.nextLine();
            String[] sArr = s.split(" ");
            int len = sArr.length;
            arr = new int[len];
            for(int i=0; i<len; i++){
                arr[i] = Integer.parseInt(sArr[i]);
            }
            // 2 7 9 3 1
            score = new int[len];
            System.out.println(Math.max(trace(len-1), trace(len-2)));
        }
    }
    public static int trace(int i){
        if(i < 0) return 0;
        if(map.containsKey(i)) return score[i];
        score[i] = Math.max(trace(i-2) + arr[i], trace(i-3) + arr[i]);
        map.put(i, score[i]);
        return score[i];
    }
}
```