

Java-玩牌高手-给定一个长度为 n 的整型数组

题目描述:

给定一个长度为 n 的整型数组，表示一个选手在 n 轮内可选择的牌面分数。选手基于规则选牌，请计算所有轮结束后其可以获得的最高的总分。选择规则如下：

1、在每轮里选手可以选择获取该轮牌面，则其总分加上该轮牌面分数，为其新的总分。

2、选手也可不选择本轮牌面直接跳到下一轮，此时将当前总分还原为 3 轮前的总分，若当前轮次小于等于 3（即在第 1、2、3 轮选择跳过轮次），则总分置为 0。

3、选手的初始总分为 0，且必须依次参加每一轮。

输入描述:

第一行为一个小写逗号分割的字符串，表示 n 轮的牌面分数， $1 \leq n \leq 20$ 。

分数值为整数， $-100 \leq \text{分数值} \leq 100$ 。

不考虑格式问题。

输出描述:

所有轮结束后选手获得的最高总分。

补充说明:

示例1
输入: 1,-5,-6,4,3,6,-2
输出: 11
说明: 总共有7轮牌面。
第一轮选择该轮牌面，总分为1。
第二轮不选择该轮牌面，总分还原为0。
第三轮不选择该轮牌面，总分还原为0。
第四轮选择该轮牌面，总分为4。
第五轮选择该轮牌面，总分为7。
第六轮选择该轮牌面，总分为13。
第七轮如果不选择该轮牌面，则总分还原到3轮1前分数，即第四轮的总分4，如果选择该轮牌面，总分为11，所以选择该轮牌面。
因此，最终的最高总分为11。

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

```
// 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);  
        String[] arr = in.nextLine().split(",");  
        int a = arr.length;  
        int[] n = new int[a];  
        for(int i=0;i<a;i++)  
        {  
            n[i] = Integer.parseInt(arr[i]);  
        }  
        int[] dp = new int[a+1];  
        dp[0] = 0;  
        for(int i=1;i<=a;i++)  
        {
```

```
        dp[i] = Math.max(dp[i-1] + n[i-1],  
        i < 3 ? 0 : dp[i-3]);  
    }  
    System.out.println(dp[a]);  
}  
}
```