

题目描述：

小明和朋友玩跳格子游戏，有 n 个连续格子组成的圆圈，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，不能回头跳，也不能超过一圈；
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，第一个格子和最后一个格子收尾相连，如：2 3 2

输出描述：

输出能够得到的最高分，如：3

补充说明：

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

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

示例 1

输入：

2 3 2

输出：

3

说明：

只能跳 3 这个格子，因为第一个格子和第三个格子收尾相连

示例 2

输入：

1 2 3 1

输出：

4

说明：

$1+3=4$

```
#include<iostream>
```

```
#include<vector>
```

```
using namespace std;
```

```
int main(){
```

```
    vector<int> nums;
```

```
    int num;
```

```
    while(cin>>num)
```

```
        nums.push_back(num);
```

```
    int len=nums.size();
```

```
    if(len<4){
```

```
        if(len==1){
```

```

        cout<<nums[0];
        return 0;
    }
    else if(len==2){
        cout<<max(nums[0],nums[1]);
        return 0;
    }
    else{
        cout<<max(max(nums[0],nums[1]),nums[2]);
        return 0;
    }
}
vector<int> num1(len,0),num2(len,0);
int ans=0;
num1[0]=nums[0],num1[1]=nums[1],num1[2]=max(nums[0]+nums[2],nums[1]);
for(int i=3;i<len-1;++i){
    num1[i]=max(max(num1[i-2]+nums[i],num1[i-3]+nums[i]),num1[i-1]);
}
num2[0]=0,num2[1]=nums[1],num2[2]=nums[2];
for(int i=3;i<len;++i){
    num2[i]=max(max(num2[i-2]+nums[i],num2[i-3]+nums[i]),num2[i-1]);
}
cout<<max(num1[len-2],num2[len-1]);
}

```