

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:

给定一个数列, 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 <bits/stdc++.h>
```

```
using namespace std;
```

```
int k;
```

```
int main()
```

```
{
```

```
    int a;
```

```
    vector<int> arr;
```

```
    while (cin >> a)
```

```
    {
```

```
        arr.push_back(a);
```

```
        if (cin.get() == '\n')
```

```
        {
```

```
            break;
```

```
        }
```

```
    }
```

```
    int result = 0;
```

```
    int n = arr.size();
```

```
    if (n == 1)
```

```

{
    cout << arr[0];
}
else
{
    if (n == 2)
    {
        cout << max(arr[0], arr[1]);
    }
    else
    {
        vector<int> b(n, 0);
        b[0] = arr[0];
        b[1] = max(arr[0], arr[1]);
        for (int i = 2; i <= n - 1; i++)
        {
            b[i] = max(b[i - 2] + arr[i], b[i - 1]);
        }

        result = b[n - 2];
        vector<int> c(n, 0);
        c[1] = arr[1];
        c[2] = max(arr[1], arr[2]);
        for (int i = 3; i <= n - 1; i++)
        {
            c[i] = max(c[i - 2] + arr[i], c[i - 1]);
        }
        result = max(result, c[n - 1]);
        cout << result;
    }
}
system("pause");
return 0;
}

```