

1418: 猴子选大王

题目描述

由经典约瑟夫问题改成。有 N 个猴子，编号从1到 N 。每个猴子对应一个正整数 X_i ，表示如果从编号为 i 的猴子开始报数，需要数到 X_i 。

这 N 个猴子围成一圈，从第一个开始报数，数到第1个猴子对应的正整数 X_1 的猴子出队，然后从它的下一位继续从1开始报数，数到对应的 X_i 时出队，如此循环直到剩下一个猴子，最后剩下的那个猴子就是猴子们选出的大王。

例如：

$N=5, X_i$ 对应为：1, 2, 3, 4, 5。

出队的顺序为：1, 3, 4, 5。

输入

第一行为 N ；

第二行为 N 个小于等于100的正整数。对应于从某个猴子位置开始报数，需要报数的次数。

输出

被选为大王的猴子的编号。

输入样例

```
5
1 2 3 4 5
```

输出样例

```
2
```

解析

依次从头报数，遇到指定的数字则出队，示例完整过程如下，红色表示出队：

数到1

索引	1	2	3	4	5
	1				

数到2

索引	2	3	4	5
	1	2		

数到4

索引	2	4	5
	3	1	2
		4	

数到5

索引	2	5
	2	1
	4	3
		5

最终剩余

索引	2

编码

```
#include <bits/stdc++.h>

using namespace std;

int n, a[1000050], cnt, m;

int main() {
    cin >> n;
    queue<int> q;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
        q.push(i); //入队
    }
    m = a[1]; // 第一个要求的数其实就是a[1]
    //一直循环直到剩余最后一人
    while (q.size() != 1) {
        ++cnt; // 报数
        int x = q.front();
        //报完数字的出队，根据后续条件判断是否重新入队
        q.pop();
        if (cnt != m)
        {
            q.push(x); // 若没到要求的数，则入队，继续报数
        }
        else { // 同循环队列
            cnt = 0;
            m = a[q.front()]; // 出队，并更换m
        }
    }
    cout << q.front() << endl;
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

