

## P1996 约瑟夫问题

### 题目描述

$n$  个人围成一圈，从第一个人开始报数，数到  $m$  的人出列，再由下一个人重新从 1 开始报数，数到  $m$  的人再出圈，依次类推，直到所有的人都出圈，请输出依次出圈人的编号。

注意：本题和《深入浅出-基础篇》上例题的表述稍有不同。书上表述是给出淘汰  $n-1$  名小朋友，而该题是全部出圈。

### 输入格式

输入两个整数  $n, m$ 。

### 输出格式

输出一行  $n$  个整数，按顺序输出每个出圈人的编号。

### 输入样例

10 3

### 输出样例

3 6 9 2 7 1 8 5 10 4

### 解析

这是一道很好的队列问题

首先我们需要模拟一个队列，将所有的元素压进队列

在进行循环（直到队列为空为止） 首先你要知道：

队列只可以在head删除，那么这就要求我们只要这个人经过判断并且不会被剔除，那么就必须把他排在队尾；

若这个人正好被剔除，那先输出他，再踢除。

### 编码

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

using namespace std;

list<int> nums;

int main() {
    //n个小朋友，数m个数
```

```

int n, m, cnt = 0;
cin >> n >> m;
//将同学们加入到链表中
for (int i = 1; i <= n; ++i) {
    nums.push_back(i);
}
//定义两个迭代器，分别代表枚举位置
list<int>::iterator it;
//待删除迭代器
list<int>::iterator toDelete;
//从头开始枚举
it = nums.begin();
//链表不为空
while (!nums.empty()) {
    cnt++;
    //备份出来一个待删除元素的指针
    //注意，此处一定要备份，否则执行删除指令后，
    //该迭代器指向的节点也就不存在了，
    //继续使用这个迭代器则会发生异常
    toDelete = it;
    //循环到结尾，重新开始
    //注意，此处依然是左闭右开的数据范式，注意判断
    //另外一定要先移动迭代器，再进行数量判断，原理同备份
    if (++it == nums.end()) {
        it = nums.begin();
    }
    //数量足够了
    if (cnt == m) {
        //打印出圈的同学编号
        cout << *toDelete << " ";
        //将这个小朋友从链表中移除
        nums.erase(toDelete);
        //重置计数器
        cnt = 0;
    }
}
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。

