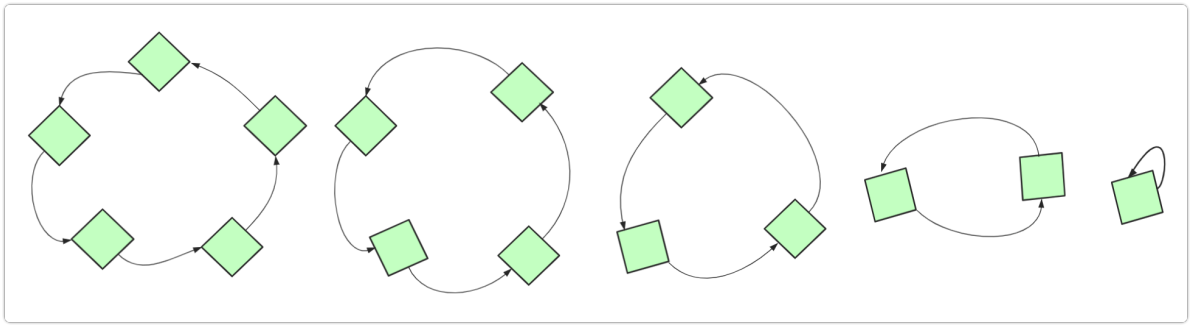
约瑟夫问题

0,1,···,n-1这n个数字排成一个圆圈，从数字0开始，每次从这个圆圈里删除第m个数字（删除后从下一个数字开始计数）。求出这个圆圈里剩下的最后一个数字。

循环链表模拟

这个问题最本质其实就是循环链表的问题，围成一个圈之后，就没有结尾这就是一个典型的循环链表嘛！一个一个顺序报数，那不就是链表的遍历枚举嘛！数到对应数字的出列，这不就是循环链表的删除嘛！



并且这里还有非常方便的地方：

- 循环链表的向下枚举不需要考虑头尾问题，直接node=node.next向下
- 循环聊表的删除也不需要考虑头尾问题，直接node.next=node.next.next删除

当然也有一些需要注意的地方

- 形成环形链表很简单，只需要将普通链表的最后一个节点的next指向第一个节点即可
- 循环链表中只有一个节点的时候停止返回，即node.next=node的时候
- 删除，需要找到待删除的前面节点，所以我们删除计数的时候要少即一位，利用前面的那个节点直接删除后面节点即可

这样，思路明确，直接开撸代码：不会一点

下标索引变换

```
#include "stdio.h"

int main() {
    int n, arr[100], m;//n是有多少个人，用arr数组来表示报数顺序，m是有多少人
    scanf("%d %d", &n, &m);
    for (int i = 0; i < n; i++) {
        arr[i] = i + 1;
    }
    int index = 0, count = 0;//用index标记出圈的人,用count标记已经出圈的人数
    int step = 0;//用step来标识每轮的计数
    while (count < n) {
        if (arr[index] != 0) { //判断这个人是否出局
            step++;
            if (step == m) {
                printf("%d ", arr[index]);
```

```
                arr[index] = 0;
                count++;
                step = 0;
            }
        }
        index = (index + 1) % n;//用取模将顺序结构变成闭环，精髓
    }

    return 0;
}
```

```
#include<stdio.h>
int main()
{
    int a[100];
    int index=0;
    int count=0;
    int n,m;
    scanf("%d%d",&n,&m);
    for(int i=0;i<n;i++)
    {
        a[i]=i+1;
    }
    while (count<n)
    {
        int step=0;
        while (step<m)
```

```
        {
        if(a[index]!=0)
        {
            step++;
        }
        index=(index+1)%n;
    }
        printf("%d",a[(index+n-1)%n]);
        a[(index-1+n)%n]=0;
        count++;
    }

    return 0;
}
```