

约瑟夫环的三种解法

约瑟夫问题是个有名的问题：N个人围成一圈，从第一个开始报数，第M个将被杀掉，最后剩下一个，其余人都将被杀掉。 N个人从1开始编号，问最后活下来的人的编号是多少。

方法1：数组模拟。

```
XXXXXXXXXX

#include<bits/stdc++.h>

using namespace std;

int main() {

    printf("-----约瑟夫环问题-----\n");

    printf("请输入总共人数和报数死亡的编号数:");

    int n,m;

    cin>>n>>m;

    bool a[n+1];

    memset(a,0,sizeof(a));

    int tot=0,cnt=0,i=0;//当前死亡总人数,报数的报数器.

    while(cnt<n) {

        i++;

        if(i>n) i=1;

        if(!a[i]) cnt++;

        if(cnt==m) {

            cnt=0;

            tot++;

            a[i]=1;

            printf("第%d个死的人的编号为%d\n",tot,i);

        }

    }

    printf("游戏结束,所有人死亡\n");

    return 0;

}
```

方法2：数学递推

主要是利用递推的思想,令 $Y[i]$ 为有 i 个人时最后活着的人的编号(从0开始编号)。显然 $Y[1]=0$,现在我们来考虑 $Y[2]$ 为多少。我们已知报到 $m-1$ 的会死. 所有当有两个人的时候.死的就是那个报 $m-1$ 的人.然后编号为 m 的那个人就活了下来.所以 $Y[2] = Y[1] + m$ 依次进行递推: 可得 $Y[i] = Y[i-1] + m$, 由于我们知道报数类似于一个循环.所以编号可能会超出范围.所以我们对每一个进行取模。 $Y[i] = (Y[i-1] + m) \% i$,对 i 取模 保证编号是从0到 $i-1$ 。

```
xxxxxxxxxx

#include<bits/stdc++.h>

using namespace std;

int main() {

    int n,m;

    while(cin>>n>>m) {

        int p=0;

        for(int i=2;i<=n;i++) {

            p=(p+m)%i;// i=2时是n=2时最后活着的人编号(从0开始), i=1时, 最后活着的人编号肯定为0

        }

        printf("最后活着的人的编号为%d\n", p+1);

    }

}
```

方法3: 队列实现.

将队伍看成是一个循环队列.如果没点到第 m 个人就放到队尾,否则弹出,cnt重置.下面上代码.

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

using namespace std;

int main() {

    queue<int>q;

    int n,m;

    cin>>n>>m;

    for(int i=1;i<=n;i++) q.push(i);

    int cnt=1;

    while(q.size())

    {

        if(cnt==m)

        {

            printf(q.size()==1?"%d\n":"%d ", q.front());

            q.pop();

        }

    }

}
```

```
        cnt=1;
    }
    else {
        int tmp=q.front();
        cnt++;
        q.pop();
        q.push(tmp);
    }
}
return 0;
}
```