题目：约瑟夫环

班级：电信1808 姓名：蔡浩晖 学号：U201813453 完成日期：2019年3月25日星期一

1. 需求分析
2. 首先输入总人数n<=30，在依次输入第二个人到第n个人的密码，最后输入第一个人的密码。然后输入初始m。输入值均为正数。
3. 输入整形
4. 计算在初始条件下的约瑟夫环的出队顺序。
5. 正确输入：7 1 7 2 4 8 4 3 6

输出结果：6 1 4 7 2 3 5

错误输入：31或者-2

输出结果：n值不符合要求。

1. 概要设计
2. 流程：先根据用户要求的节点数创建一个循环链表，再将编号及密码录入。

然后通过用户给的初始化的m值，开始程序。最后得出最终结果。

1. 主函数调用建立链表函数。
2. ADT josephe{

数据对象：List p, int a[]，int n,int m;

数据关系：n>0,m>0

基本操作：

Init（）

建立链表

Createjosephe(List p,int \*a)

运行约瑟夫环

}ADT josephe

1. 详细设计
2. 伪码同上
3. Init（）

Josephe（）

属于 main（）

1. 调试分析
2. 遇到最后两个值相反

解决办法：最后m-2为负数，因此加上绝对值即可解决

1. 时空分析

Init（） 为O（1）；

Josephe（）为O（n）

1. 用户使用说明

先输入n，在依次输入密码，最后输入第一个密码

1. 输入结果

输入：7 1 7 2 4 8 4 6

输出：6 1 4 7 2 3 5

输入：6 1 2 3 4 5 6 6

输出：6 5 4 3 2 1

1. 附录

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

typedef struct Node\* List;

struct Node{

int data;

int number;

List next;

int \* Joseph(List p,int \*a,int n);

};

List init()

{

List L =(List)malloc(sizeof(Node));

if(L==NULL)

{

printf("内存不足");

}

L->next = NULL;

return L;

}

int \*Joseph(List p,int \*a,int n)

{

List head,\_L;

head = p;

int k,m;

int i;

printf("Input the m\n");

scanf("%d",&m);

i = 0;

while(head->next!=head)

{

for(k=0;k<abs(m-2);k++)

head = head->next;

a[i]=head->next->number;

m = head->next->data;

\_L = head->next;

head->next = head->next->next;

head = head->next;

i++;

free(\_L);

}

a[n-1] = head->number;

return a;

}

int main(void)

{

int n;

scanf("%d",&n);

if(n>20||n<0)

{

printf("Input error");

return 0;

}

int i,j,k,m;

j = n;

int \*a = (int\*)malloc(n\*sizeof(int));

List p,head,L,\_L;

p = init();

head = p;

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

{

L = init();

L->number = i;

scanf("%d",&L->data);

if(L->data<0)

printf("error");

p->next = L;

p = L;

}

p->next = head;

head->number = 1;

printf("Input the first data\n");

scanf("%d",&head->data);

a = Joseph(head,a,n);

for(i=0;i<n;i++)

{

printf("%d ",a[i]);

}

}