约瑟夫环 电信1808 连琛 U201813455 2019.3.25

约瑟夫环问题的一种描述是：编号为1,2，…，n的n个人按顺时针方向围坐一圈，每人持有一个密码（正整数）。一开始任选一个正整数作为报数上限值m，从第一个人开始按顺时针方向自1开始顺序报数，报到m时停止报数。报m的人出列，将他的密码作为新的m值，从他在顺时针方向上的下一个人开始重新从1报数，如此下去，直至所有人全部出列为止。试设计一个程序求出出列顺序。

1. 需求分析

程序的任务是求出出列的人的编号顺序，依次输入n，a[i],m，均为整型变量，0<n<100,a[i]>0,m>0。输出为依次出列的人的编号。要求程序能正确运行并计算出正确结果。

正确输入

7

3 1 7 2 4 8 4

20

错误输入

5

-1 0 4 2 8

-3

2.概要设计

typedef struct node{

int k,x;

struct node \*next;

}Linklist;

Linklist为链表，主程序中先建立单循环链表，再从链表的头结点依次向后数m，输出编号，删除结点，直到链表为空。

3. 定义以及输入

int a[100],i,m,n,t=0;

scanf("%d",&n);

for (i=0;i<n;i++) scanf("%d",&a[i]);

scanf("%d",&m);

Linklist \*l,\*p,\*q,\*s;

创建单循环链表

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

{

p=(Linklist\*)malloc(sizeof(Linklist));

p->k=a[i]; p->x=i+1;

if (i==0) l=p;

else {q->next=p;}

q=p;

}

q->next=l; p=l;

进行运算

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

{

p=(Linklist\*)malloc(sizeof(Linklist));

p->k=a[i]; p->x=i+1;

if (i==0) l=p;

else {q->next=p;}

q=p;

}

q->next=l; p=l;

while (t<n)

{

for (i=0;i<m-1;i++) { p=p->next; q=q->next;}

printf("%d\t",p->x);

m=p->k; q->next=p->next; free(p); p=q->next;

t++;

}

4.调试分析

时间复杂度为∑a[i],空间复杂度为o（1）。

5.用户使用说明

拷入codeblocks中可直接运行。

6．测试结果

输入

7

3 1 7 2 4 8 4

20

输出

6 1 4 7 2 3 5

输入

5

1 8 3 4 5

13

输出

3 1 2 5 4