# 约瑟夫环

电信1808曹子龙U201813454

2019/3/27

该程序的目的是利用单向循环链表存储结构模拟约瑟夫问题：依次输入编号为1，2，3……，n，共n（n<=30）个人的密码（整型）和上限值m（整型），从第一个人开始由1开始顺序报数，将报m的数出列并将他的密码作为新的m值重新开始报数……并按照出列的顺序打印各人的编号。

【测试数据】m=20，n=7，7个人的密码依次为：3，1，7，2，4，8，4，正确的出列顺序为6，1，4，7，2，3，5。

【概要设计】

定义结点储存每个人的信息代表每个人的密码和序号。

Loop\* creat()函数创建单项循环链表存储结点并输入数据。

void Joseph()函数进行报数和出列操作。

在主函数中调用creat函数创建链表并输入；输入上限值m；调用Joseph函数进行报数和出列。

【详细设计】

定义结点：

typedef struct Joseph

{

int password;

int number;

struct Joseph\* next;

} Loop;

创建链表：

creat(int m)

{

Loop\* head=(Loop\*)malloc(sizeof(Loop));

Loop \*p,\*e;

if(!head) return 0;

int a;

printf("第1人的密码：");

scanf("%d",&a);

head->password=a;

head->number=1;

head->next=NULL;

p=head;

int x;

for(int i=1;i<m;++i)

{

if(!(e=(Loop\*)malloc(sizeof(Loop)))) return 0;

printf("第%d人的密码：",i+1);

scanf("%d",&x);

e->number=i+1;

e->password=x;

p->next=e;

p=e;

}

p->next=head;

return p;

}

报数操作：

Joseph(Loop\* p,int m,int x)

{

Loop\*q;

int i;

if(x==0) return 0;

q=p;

m%=x;

if(m==0) m=x;

for(i=1;i<=m;++i)

{

p=q;

q=p->next;

}

p->next=q->next;

i=q->password;

printf("%d\n",q->number);

free(q);

Joseph(p,i,x-1);

}

【调试分析】

creat函数时间复杂度O(n)；Joseph函数的时间复杂度O(n)；

本程序的creat和Joseph函数仍有精简的空间。

【使用说明】

在程序运行前可在头文件#define n 中改变n的值；运行程序，根据提示依次输入n个人的密码和上限值m，得到输出结果。

【测试结果】

n=7 输入：3，1，7，2，4，8，4 m=20 输出：6，1，4，7，2，3，5；

n=7 输入：1，2，3，4，5，6，7 m=23 输出：2，4，1，3，7，5，6；

【附录】

Joseph1