实习报告：1.2题 约瑟夫环

题目：编制一个模拟约瑟夫问题的程序

班级：电信1809 姓名：文竞宇 学号：U201813487 完成日期：2019.3.25

1. 需求分析
2. 本程序要求输入人数、初始报数上限和各个人的密码，输入均为int型。程序会自动输出出局的编号顺序。
3. 程序以用户和计算机对话的方式执行，即在计算机终端上显示“提示信息”后，由用户在键盘上输入程序中相应的输入数据，运算结果显示在其后。
4. 程序执行的命令包括：1）构造约瑟夫环；2）计算出局顺序；3）结束。
5. 测试数据：

输入：

m = 20

n = 7

3 1 7 2 4 8 4

输出：

6 1 4 7 2 3 5

1. 概要设计

为实现上述程序功能，应以循环链表表示约瑟夫环。

1. 抽象数据类型循环链表的定义如下：

**ADT CLinkList** **{**

**基本操作：**

InitList(&L)

操作结果：构造一个空的循环链表。

后面的不会写了。

**} ADT CLinkList**

1. 主程序

void main(){

初始化；

输入约瑟夫环初始值；

输出测试结果；

}

1. 详细设计
2. 约瑟夫环类型

typedef struct CLNode

{

datatype No;

datatype code;

struct CLNode \*next;

}CLNode, \*CLinkList;

1. 元素类型

typedef int datatype;

1. 求约瑟夫问题的伪码算法：

Status Solve(CLinkList L, int m)

{

CLinkList p=L->next, q=L;

while (p->next != p)

{

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

{

q = p;

p = p->next;

}

printf("%d ", p->No);

m = p->code;

q->next = p->next;

free(p);

p = q->next;

}

printf("%d\n", p->No);

free(p);

return OK;

}//Solve

1. 主函数和其他函数的算法

CLinkList InitNode()

{

CLinkList L = (CLinkList)malloc(sizeof(CLNode));

L->next = NULL;

return L;

}

void CreateCL(CLinkList head, int n)

{

CLinkList tail, p;

tail = head;

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

{

p = InitNode();

p->No = i;

printf("No.%d Code=", i);

scanf\_s("%d", &p->code);

tail->next = p;

tail = p;

}

tail->next = head->next;

}

int main()

{

int m, n;

printf("m=");

scanf\_s("%d", &m);

printf("n=");

scanf\_s("%d", &n);

CLinkList L=InitNode();

CreateCL(L, n);

if (m%n)

m %= n;

else

m = n;

Solve(L, m);

return 1;

}

1. 调试分析
2. 在定义循环链表的过程中一开始没有使用尾指针，导致构造链表的算法很复杂。
3. 在设计删除结点的算法时利用了头指针，可以保证删去每一个对象。
4. 本程序没有讨论当输入的值并非正整数时候的返回值，需要完善。
5. 算法的时空分析

InitNode是O(1)的，CreateCL，Solve是O(n)的，n为链表长度。

1. 用户手册（略）
2. 测试结果（略）
3. 附录

源程序文件名清单：

Training 1 Joseph Circle.cpp //主程序