题目：约瑟夫环

班级：电信1808班

姓名：郭佳璐

学号：U201813461

完成日期：2019年3月25日星期一

1. **需求分析：**

创建一个有n个元素的无头结点的单循环链表，程序于编号为1的元素开始向后计数至m时，将该元素的编号输出，m值变为该元素的值并删除该数，并从该数的下一个数开始报数，直至计数至m时，重复上述步骤直至链表为空。

按提示输入n+1个数，均为整型；输出以整型输出。

程序构造了一个单循环链表，并能够按照题目要求让元素以某种顺序出列。

**2.概要设计**：

**基本操作：**

link\_create(n)

操作结果：创建一个含有n个元素的无头结点单链表

link\_process(head)

初始条件：链表Node已存在。

操作结果：按题目要求有序输出元素。

**本程序包含四个模块：**

1. 主程序模块：

int main(){

初始化；

接受命令；

处理命令；

}

1. 节点结构单元模块——定义链表的节点结构。
2. 链表单元模块——实现链表元素的赋值。
3. 元素出列单元模块——实现元素编号的出列。

**3.详细设计**：

1.元素类型，节点类型和指针类型

typedef struct node{

int num;

int data;

struct node \*next;

}Node;

1. 根据有序表的基本操作的特点，有序表采用无头结点的单向循环链表实现，链表设一个头指针和一个尾指针，利用尾指针指向头指针实现链表的循环。

Node \*link\_create(int n)

{

Node \*p,\*q,\*head;

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

head=p; //\*p,\*head指向第一个元素节点

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

{

q=(Node \*)malloc(sizeof(Node));

q->num=i;

p->next=q;

p=q； //\*p,\*q指向最后一个元素节点

}

p->next=head; //使得单向链表变为单循环链表

return head;

}

1. 利用循环链表的特性实现有序输出各元素编号

void link\_process(Node \*head)

{

int i,m;

printf("input the m:");

scanf("%d",&m);

Node \*p=head;

Node \*tmp1;

while(p->next!=p)

{

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

{

tmp1=p;

p=p->next;

}

printf("%5d",p->num);

m=p->data;

tmp1->next=p->next;

free(p);

p=tmp1->next;

}

printf("%5d",p->num);

free(p);

}

**4.调试分析：**

调试过程中，发现当m=6时，得到的第一个元素的编号为7，经分析发现是由于初始循环时i的值赋为0，使得指针多向前移了一位，将0改为1后，程序结果正确。

**5.用户使用说明：**

可通过在主函数中直接更改m的值和n的值，运行程序，根据提示在键盘上输入各元素

对应的密码值，即可得到结果。

**6.测试结果：**

input the n:7

input the 1th data:3

input the 2 th data:1

input the 3 th data:7

input the 4 th data:2

input the 5 th data:4

input the 6 th data:8

input the 7 th data:4

input the m:6

输出结果：

6 1 4 7 2 3 5

1. **附录**