**“约瑟夫环”实验报告**

题目：数据结构题集p79的“约瑟夫环”

班级：电信1808

姓名：瞿博文

学号：U201813460

日期：2019年3月25日星期一

1. 需求分析
2. 程序设计的任务：对编号1，2，....，n的n个人按顺时针方向围坐一圈，每个人持有一个正整数的密码。一开始任选一个正整数作为报数上限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。通过程序可以得到每一次出列的人的编号和最后出列的人的编号。
3. 输入的形式：通过scanf函数进行键盘输入

输入值的范围：m为正整数，n为大于1的正整数

1. 输出的形式：通过printf进行输出
2. 程序所达到的功能：可以得到每一次出列的人的编号和最后出列的人的编号
3. 测试数据：

输入：m的初值为20，n为7，密码分别为：3，1，7，2，4，8，4

输出：

出队列的是6号

出队列的是1号

出队列的是4号

出队列的是7号

出队列的是2号

出队列的是3号

最后出列的是5号

输入：m为20，n为10，密码为4 5 1 7 2 8 9 1 2 5

输出：出队列的是10号

出队列的是5号

出队列的是7号

出队列的是9号

出队列的是2号

出队列的是1号

出队列的是8号

出队列的是3号

出队列的是4号

最后出列的是6号

1. 概要设计

ADT：循环单链表

主程序：主函数进行输入m和n的值并构建循环单链表，并在主函数中构建循环单链表并对每个结点的编号和密码进行赋值和输入

各程序模块：1.先定义结点（结构体）2.再用主函数进行m和n的值的输入3.在主函数中构建循环单链表并对每个结点的编号和密码进行赋值和输入4.主函数调用josephu函数来进行对每次出列的人的序号和最后一次出列的人的序号的计算

三：详细设计

#include<stdio.h>

#include<stdlib.h>

typedef struct node

{

int num;

int data;

struct node \*next;

}node;

void josephu(node \*head,int m);

void main()

{

node \*p=NULL;

node \*s=NULL;

node \*head=NULL;

int n;

printf("please input the number of people : n= \n");

scanf("%d",&n);

int m;

printf("please input the max out number : m= \n");

scanf("%d",&m);

int\* a;

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

printf("please input the data \n");

for(int j=0;j<n;j++)

scanf("%d",&a[j]);

int i;

for(i=1;i<=n;i++) //构建循环链表

{

s=(node\*)malloc(sizeof(node));

s->num=n-i+1;

s->data=a[n-i];

s->next=NULL;

if(i==1)

{

p=s;

}

s->next=head;

head=s;

}

p->next=head;

josephu(head,m);

}

void josephu(node \*head,int m)

{

int i;

node \*p=NULL;

node \*q=NULL;

p=head;

while(p->next!=p)

{

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

{

q=p;

p=p->next;

}

printf("出队列的是%d号\n",p->num);

m=p->data;

q->next=p->next;

p=q->next;

}

printf("最后出列的是%d号\n",p->num);

}

四．调试分析

（1）：问题1：如何构建循环单链表

解决：类比构建普通单链表（从头到尾构建），而循环单链表每个结点是等价的，通过malloc结构体的指针从尾到头即可构建出循环单链表

问题2：密码要通过数组来进行赋值，而n是变量，怎样定义这样的数组

解决：通过malloc一个int类型的指针a，大小是sizeof（int）\*n，则指针a就相当于指针的首地址，相当于构建了一个长度为n的int型数组

问题3：第一次的输出结果不正确，原因在哪？

答：第一次对结点的编号和密码赋值的时候，没有注意构建循环单链表的时候是从尾到头构建的，这样之后每次给malloc指针赋值时应该用a【n-i】（i=1；i<n;i++)而不是a【i-1】，对序号赋值的时候应该也应该是n-i+1而不是i。

1. 时间复杂度和空间复杂度分析

时间复杂度：O（n\*n） 空间复杂度：O（n）

1. 经验和体会
2. 对构建循环单链表更加了解和熟悉
3. 对删除结点直接改动指针的操作更加熟练
4. 对赋值操作和逻辑的相关性理解更加优秀
5. 用户使用说明

步骤一：定义、构建结点，对josephu函数进行声明

步骤二：在主函数中对m和n的值进行输入，并malloc一个int型指针，从而得到int型数组，进行密码的输入

步骤三：构建循环单链表同时对每个结点的编号和密码值进行赋值

步骤四：调用josephu函数，传入m的值和头指针（head）

步骤五：在被调用函数中进行“报数”和“出列”（删除）操作

步骤六：输出结果即可

1. 测试结果

输入：m的初值为20，n为7，密码分别为：3，1，7，2，4，8，4

输出：

出队列的是6号

出队列的是1号

出队列的是4号

出队列的是7号

出队列的是2号

出队列的是3号

最后出列的是5号

输入：m为20，n为10，密码为4 5 1 7 2 8 9 1 2 5

输出：出队列的是10号

出队列的是5号

出队列的是7号

出队列的是9号

出队列的是2号

出队列的是1号

出队列的是8号

出队列的是3号

出队列的是4号

最后出列的是6号