实习报告

**题目：约瑟夫环**

**班级：电信1808 姓名：文一晴 学号：U201813452 完成日期：2019.3.25**

1. 需求分析
2. 程序运行后，要求用户指定初始报数上限值m，然后读取个人密码。设n（人数）≤30，按照出列顺序打印个人编号
3. 程序执行的命令包括：
4. 构造结构体
5. 构造循环链表
6. 循环遍历链表
7. 删除节点
8. 重复（3）（4）直至最后一个节点
9. 结束
10. 测试数据

n=7，m初值为6，个人密码为：3,1,7,2,4,8,4，出列顺序：6,1,4,7,2,3,5

1. 概要设计

为实现上述功能，应以单向循环链表模拟此过程。

1. 链表的抽象数据类型定义为：

ADT list{

数据对象：D={ai|ai∈stu，i=1,2，…n，n≥0}

基本操作：

Setlist（&head）

操作结果：建立一个循环链表

Deletelink（&p）

初始条件：链表L存在

操作结果：删除一个节点

Josephloop（&head）

初始条件：链表L存在

操作结果：打印出个人出列顺序

}ADT Josephloop

1. 本程序包含四个模块：
2. 主函数模块：

Int main（）{

提示用户输入信息

释放节点空间

}

1. 初始化链表模块——实现链表的抽象数据类型
2. 约瑟夫环模块——实现约瑟夫操作
3. 删除节点模块——实现删除节点操作
4. 详细设计
5. 节点

typedef struct student

{

int ID;

int number;

struct student \*next;

}stu;

2.

stu \*set(int n,stu \*head)

{

int j;

stu \*p=head;

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

{

stu \*stup=(stu \*)malloc(sizeof(stu));

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

scanf("%d",&stup->ID);

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

scanf("%d",&stup->number);

if(head==NULL)

{

head=stup;

p=stup;

}

else

{

p->next=stup;

p=p->next;

}

}

p->next=head;

return head;

}

3.stu \*deletelink(stu \*p)//删除传入节点的下一个

{

stu \*q=p;

stu \*s;

s=q->next;

q->next=s->next;

free(s);

return q;

}

4. stu \* josephloop(stu \*head,int m,int n)

{

stu \*p=head;

int i=0;

int newm=m-1;

int idnumber;

while(n!=1)

{

for(i=0;i<newm-1;i++)

p=p->next;

newm=p->next->number;

idnumber=p->next->ID;

p=deletelink(p);

printf("%d->",idnumber);

n--;

}

idnumber=p->ID;

printf("%d\n",idnumber);

return p;

}

5. int main()

{

int n,m;

printf("please input the number of student\n");

scanf("%d",&n);

printf("please input the first number\n");

scanf("%d",&m);

stu \*head=NULL;

head=set(n,head);

head=josephloop(head,m,n);

free(head);

}

1. 调试分析

时空分析：本程序空间复杂度由人数n决定，空间复杂度为O（n）

1. 测试结果

输入n=7，m=6，并初始化个人密码分别为：3,1,7,2,4,8,4

出列顺序：6->1->4->7->2->3->5

1. 附录

源程序文件名清单

Stu //节点名

Set //建立链表

Deletelink //删除节点

Josephloop //约瑟夫环实现单元