

学号: _____ 姓名: _____ 班级: _____ 成绩: _____

实验地点：理学院机房

一、实验目的:

- ## 二、评分标准:

- 三、上机实验前完成下述问题:

- ### 1、线性结构的 4 个基本特征是什么？

答：

- ## 2、线性表的顺序表示指的是什么？

答：

3、线性表的链式存储结构的特点是什么？

答：

4、线性表的顺序存储和链式存储的优缺点是什么？

答：

5、单链表的优缺点是什么？

答：

6、循环链表的特点是什么？

答：

7、 双向链表的操作特点是什么？

答：

四、 算法操作

所要实现的程序：设有 n 个人围坐在一圈，现从指定的第 1 个人开始报数，数到第 m 个人出列，然后从出列的下一个人重新开始报数，数到第 m 个人又出列，如此重复，直到所有的人全部出列为止。

算法的参考思想

通过我们对约瑟夫问题的分析，我们认为单循环链表能较好的解决问题。在建立循环链表时，因为约瑟夫环的大小由输入决定。因此与匹配的结点数也是变化的，所以要进行动态内存分配。输出数据的存储单元也就需要动态分配了，为方便操作，我们将每个结点数据域的值定为生成该结点时的顺序号。

进行操作时，我们用一个变量 j 来记住扫描过的结点，当扫描到需输出的结点的前一个结点时，我们将其直接后继结点的数据域值输出，并释放已输出数据域值的那个结点，然后将 j 置 0，让 j 从被删除结点的下一个结点重新记数，这样不断重复这一系列动作，就可以轻松解决约瑟夫环问题。

参考源程序（给出代码后面标志为 “`/**`” 符号后进行解释说明本行代码实现的功能）

```

#include<stdio.h>
typedef struct node{          /*定义单循环链表的结构* （以下按这样的结构描述）
    int data;
    struct node*next;}   Lnode;
Lnode*create(int n)
{int i;
    Lnode*h,*p,*r=(Lnode*)malloc(sizeof(Lnode));    /* *
    _____
    r->data=n;h=r;    /* *
    _____

    for(i=n-1;i>0;i--)
        {p=(Lnode*)malloc(sizeof(Lnode)); /* *
        _____

        p->data=i;
        p->next=h;
        h=p;
        }
        r->next=h;    return h;
    }
void jeseeph(Lnode*p,int m)
{Lnode*q; int j=0;
    printf("outqueue order:");
do{
    j++;
    if(j==m-1)
        {q=p->next;    /* *
        _____

        p->next=q->next;    /* *
        _____

        printf("%d  ",q->data);
        j=0;free(q);}    /* *
        _____

        p=p->next;
    }while(p->next!=p)
    printf("%d\n",p->data);
    free(p);
}

Void main()
    {Lnode * h; Int m,n;
    printf("\n input n,m=");
    scanf("%d,%d",&n,&m);
    h=create(n);
    jeseeph(h,m);
}

```

评语:

任课教师：赵宏庆 成绩： _____ _____年__月__日