

## ① 当前作业

[» 21级第七次作业 \(图\)](#)[» 21级第六次作业 \(查找与排序\)](#)[» 21级第五次作业 \(树\)](#)[» 2021级 \(信息大类\) 数据结构综合作业 \(正确性和性能\)](#)[» 2021级 \(信息大类\) 数据结构综合作业 \(可扩展性\)](#)

## ② 历史作业

[» 21级第四次作业 \(栈和队\)](#)[» 21级第三次作业](#)[» 21级第二次作业](#)[» 21级第一次作业](#)

## 21级第三次作业 作业时间：2022-03-24 18:00:00 至 2022-04-21 23:55:00

第三次作业主要考查对顺序表和链表知识的掌握情况，请用相关知识完成本次作业。

选择填空题：20分，编程题100分，总分120分。

总分: 120.00

## -pencil 选择题

1. 首次提交时间:2022-03-24 19:38:15 最后一次提交时间:2022-03-24 19:38:15 得分: 1.00

在一个单向循环链表中，若要在p所指向的结点之后插入一个新结点，则需要相继修改  
b \_\_\_\_\_ 【 正确答案: B】 个指针域的值

- A. 1
- B. 2
- C. 3
- D. 4

2. 首次提交时间:2022-03-24 19:39:05 最后一次提交时间:2022-03-30 22:07:50 得分: 1.00

在一个以 h 为头节点的单循环链表中， p 指针指向链尾节点的条件是

A \_\_\_\_\_ 【 正确答案: A】 。

- A. p->next = h
- B. p->next = NULL
- C. p->next->next = h
- D. p->data = -1

3. 首次提交时间:2022-03-24 19:39:45 最后一次提交时间:2022-03-24 19:39:45 得分: 1.00

下面说法错误的是 (c \_\_\_\_\_ 【 正确答案: C】 )

- A. 线性关系是指：除了第一个元素与最后一个元素外，元素序列中任何一个元素有且仅有一个直接前驱元素，并且有且仅有一个直接后继元素。
- B. 数据元素之间具有的逻辑关系为线性关系的数据元素集合称为线性表。
- C. 线性表的有序性是指其中的数据元素是按照由小到大或由大到小的顺序排列的。
- D. 线性表采用顺序存储结构时，读取其中某个位置的元素的时间复杂度为O(1)。

4. 首次提交时间:2022-03-24 19:40:08 最后一次提交时间:2022-03-24 19:40:08 得分: 1.00

下列程序片段的时间复杂度是 c \_\_\_\_\_ 【 正确答案: C】

```
count=0;
for (k=1;k<=n;k*=2)
    for (j=1;j<=n;j++)
        count++;
```

- A. O(log<sub>2</sub>n)
- B. O(n)
- C. O(nlog<sub>2</sub>n)
- D. O(n<sup>2</sup>)

5. 首次提交时间:2022-03-24 19:40:32 最后一次提交时间:2022-03-24 19:40:32

得分: 1.00

**已提交**

将下列复杂度由小到大重新排序: b \_\_\_\_\_ 【正确答案: B】

1.  $n^*\log_2(n)$ , 2.  $n+n^2+n^3$ ,  
 3.  $2^4$ , 4.  $n^{0.5}$   
 A. 3,4,2,1 B. 3,4,1,2 C. 4,3,2,1 D. 4,3,1,2

6. 首次提交时间:2022-03-24 19:41:17 最后一次提交时间:2022-03-24 19:41:47

得分: 1.00

**已提交**在一个具有n个链结点的线性链表中查找某一个链结点, 若查找成功, 需要平均比较  
c \_\_\_\_\_ 【正确答案: C】个链结点。

- A. n B.  $n/2$  C.  $(n+1)/2$  D.  $(n-1)/2$

7. 首次提交时间:2022-03-24 19:42:16 最后一次提交时间:2022-03-24 19:42:16

得分: 1.00

**已提交**

数据的存储结构通常有d \_\_\_\_\_ 【正确答案: D】。

- A. 顺序存储结构和链式存储结构  
 B. 顺序存储结构、链式存储结构和索引结构  
 C. 顺序存储结构、链式存储结构和散列结构  
 D. 顺序存储结构、链式存储结构、索引结构和散列结构

8. 首次提交时间:2022-03-24 19:43:38 最后一次提交时间:2022-03-24 19:43:38

得分: 1.00

**已提交**若长度为n的线性表采用顺序存储结构, 在其第i个位置插入一个新元素的算法的时间复杂度为(c \_\_\_\_\_ 【正确答案: C】) ( $1 \leq i \leq n+1$ )。

- A. O(0) B. O(1) C. O(n) D. O( $n^2$ )

9. 首次提交时间:2022-03-24 19:48:08 最后一次提交时间:2022-03-24 19:48:09

得分: 1.00

**已提交**下面关于线性表的叙述中, 错误的是哪一个? (b \_\_\_\_\_ 【正确答案:  
B】)

- A. 线性表采用顺序存储, 必须占用一片连续的存储单元。  
 B. 线性表采用顺序存储, 便于进行插入和删除操作。  
 C. 线性表采用链接存储, 不必占用一片连续的存储单元。  
 D. 线性表采用链接存储, 便于插入和删除操作。

10. 首次提交时间:2022-03-24 19:48:47 最后一次提交时间:2022-03-30 22:14:45

得分: 1.00

**已提交**

某线性表中最常用的操作是在最后一个元素之后插入一个元素和删除第一个元素, 则采用d \_\_\_\_\_ 【正确答案: D】存储方式最节省运算时间。

- A. 单链表  
 B. 仅有头指针的单循环链表  
 C. 双链表  
 D. 仅有尾指针的单循环链表

**填空题**

1. 首次提交时间:2022-03-24 19:53:25 最后一次提交时间:2022-03-24 19:53:25

得分: 1.00

**已提交**20人从1到20编号围成一圈, 从1开始, 进行1、2报数, 报到2的人出列, 剩余的人继续从出列人的下一个人报数, 则最后剩下的人的编号为\_\_\_\_\_ 【正确  
答案: 9】。

2. 首次提交时间:2022-03-24 19:53:59 最后一次提交时间:2022-03-24 19:54:06 得分: 1.00

**已提交**

假设 $n$ 为2的乘幂，并且 $n > 2$ ，试求下列算法的时间复杂度（以 $n$ 的函数形式表示，在本课程作业和考试范围内，除非特别声明，都采用大O表示法来表示时间复杂度，并且对数都是以2为底，即 $\log_2 n$ 可表示为： $\log n$ 或者 $\log(n)$ ； $m \cdot n$ 的形式可表示为 $mn$ ）。

```
int time (int n) {
    int count=0, x=2;
    while ( x < n/2 ) {
        x=2*x;
        count++;
    }
    return (count);
}
```

时间复杂度： $O(\log n)$  【**正确答案:  $O(\log n)$  或  $O(\log(n))$  或  $O(\log(n))$** 】

3. 首次提交时间:2022-03-24 19:54:21 最后一次提交时间:2022-03-24 19:54:44 得分: 1.00

**已提交**

下面程序段的时间复杂度为 $O(nm)$  【**正确答案:  $O(mn)$  或  $o(mn)$  或  $O(mn)$  或  $O(nm)$  或  $o(nm)$  或  $O(m^*n)$  或  $O(n^*m)$** 】（采用大O表示法）。

```
for ( i = 0; i < n; i++ )
    for ( j = 0; j < m; j++ )
        a[i][j] = ( i > j ) ? ( i*i*i ) : ( j*j*j );
```

4. 首次提交时间:2022-03-24 19:56:07 最后一次提交时间:2022-03-24 19:56:42 得分: 1.00

**已提交**

设 $n$ 为大于1的正整数，计算机执行下面的语句时，带#语句的执行次数为 $n$  【**正确答案:  $n$** 】。

```
i=1;
j=0;
while(i+j<=n){
    # if(i>j)
        j++;
    else
        i++;
}
```

5. 首次提交时间:2022-03-24 19:59:04 最后一次提交时间:2022-03-24 20:01:55 得分: 1.00

**已提交**

设list<sub>a</sub>, list<sub>b</sub>分别为两个有序链表(升序)的第1个链结点的指针, 将这两个有序链表合并为一个有序链表, 并设合并后的链表的第一个链结点的指针为list<sub>c</sub>.

```
LinkList MERGELIST(LinkList lista, LinkList listb)
{
    LinkList listc, p=lista, q=listb, r;
    if(lista->data <= listb->data){
        listc=lista;
        r=lista;
        p=lista->link;
    }
    else{
        listc=listb;
        r=listb;
        q=listb->link;
    }
    while(p!=NULL&&q!=NULL){
        if(p->data <= q->data){
            r->link=p; 【 正确答案: r->link=p;】
            r=p;
            p=p->link; 【 正确答案: p=p->link; 或 p=r->link;】
        }
        else{
            r->link=q; 【 正确答案: r->link=q;】
            r=q;
            q=q->link; 【 正确答案: q=q->link; 或 q=r->link;】
        }
    }
    r->link=p!=NULL 【 正确答案: p 或 q==NULL 或 p!=NULL 或 (q==NULL) 或 (p!=NULL) 】 ?p:q;
    return listc;
}
```

6. 首次提交时间:2022-03-24 20:03:19 最后一次提交时间:2022-03-24 20:03:21 得分: 1.00

**已提交**

在一个长度为n的顺序表中第i个元素( $1 \leq i \leq n$ )之前插入一个元素时, 需向后移动 $n-i+1$  【 正确答案:  $n-i+1$  或  $n+1-i$  或  $1+n-i$  】 个元素。

7. 首次提交时间:2022-03-24 20:03:36 最后一次提交时间:2022-03-24 20:03:42 得分: 1.00

**已提交**

线性表中的插入、删除操作, 在顺序存储方式下平均移动近一半的元素, 时间复杂度为 $O(n)$  【 正确答案:  $O(n)$  或  $O(n)$  或  $O(n)$  】; 而在链式存储方式下, 若要在某个结点后插入和删除一个结点, 其时间复杂度都是 $O(1)$  【 正确答案:  $O(1)$  或  $O(1)$  或  $O(1)$  】。

8. 首次提交时间:2022-03-24 20:04:02 最后一次提交时间:2022-03-24 20:04:04 得分: 1.00

**已提交**

若某线性表采用顺序存储结构, 每个元素占4个存储单元, 首地址为100, 则第10个元素的存储地址为136 【 正确答案: 136】。

9. 首次提交时间:2022-03-24 20:04:52 最后一次提交时间:2022-03-30 22:11:06 得分: 1.00

已提交

非空线性链表中，若要在由p所指的链结点后面插入新结点q，则应执行语句\_\_\_\_  
 $q->link=p->link;$  【正确答案:  $q->link=p->link;$ 】 和\_\_\_\_  
 $p->link=q;$  【正  
 确答案:  $p->link=q;$ 】。

结点结构为

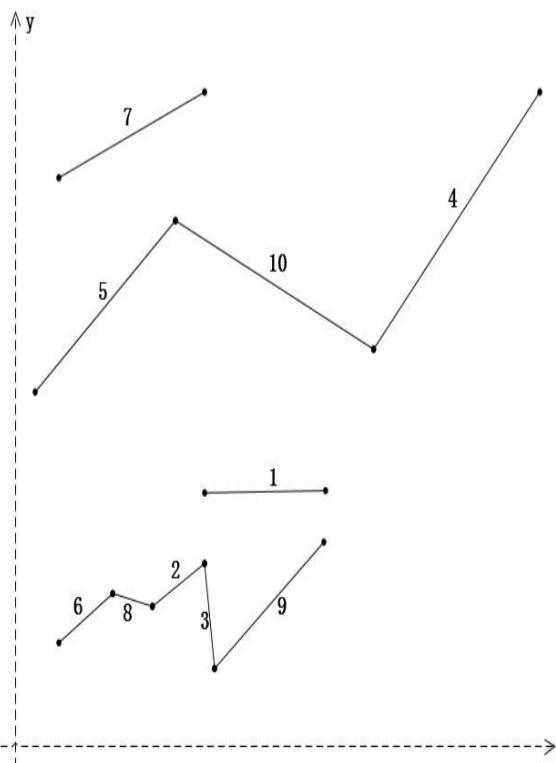
```
typedef struct Node{
    int value;
    struct Node* link;
}node;
```

10. 首次提交时间:2022-03-24 20:06:52 最后一次提交时间:2022-03-24 20:06:54 得分: 1.00

已提交

线性表 $L = (a_1, a_2, \dots, a_n)$  用数组表示，假定删除表中任一元素的概率相同，则删除一个元素平均需要移动元素的个数是\_\_\_\_ $(n-1)/2$  【正确答案:  $(n-1)/2$  或  $(n-1) / 2$ 】。

### ✍ 编程题

#	题目	分值	批阅信息											
1.	<a href="#">连续线段</a>	20.00	<a href="#">下载源文件</a>											
	<p><b>【问题描述】</b>          平面上两个点（一个点由(x,y)坐标组成）可构成一个线段，两个线段如果有一个端点相同，则可构成一个连续线段。假设构成线段的两个端点为<math>v_1(x_1, y_1)</math>和<math>v_2(x_2, y_2)</math>，在此满足<math>x_1 &lt; x_2</math>，其中<math>v_1</math>称为线段的起点，<math>v_2</math>为线段的终点。同样，对于连续线段来说，在此满足<math>x_i &lt; x_{i+1}</math> (<math>i=1 \dots n-1</math>, <math>n</math>为连续线段中的端点数，<math>x_i</math>为相应端点的X轴坐标)。输入一组线段（用两个端点的x、y坐标表示线段，<b>线段个数大于等于2，小于等于100</b>），编程计算出连续线段中包含最多线段数的线段，输出相应的线段数和起点位置（注意，<b>不是最长的连续线段，是包含最多线段的连续线段</b>）。例如：</p> 	<p>得分20.00 最后一次提交时间:2022-03-24 20:26:52          共有测试数据:5          平均占用内存:1.226K 平均CPU时间:0.00497S 平均墙钟时间:0.00494S</p> <table border="1"> <thead> <tr> <th>测试数据</th> <th>评判结果</th> </tr> </thead> <tbody> <tr> <td>测试数据1</td> <td>完全正确</td> </tr> <tr> <td>测试数据2</td> <td>完全正确</td> </tr> <tr> <td>测试数据3</td> <td>完全正确</td> </tr> <tr> <td>测试数据4</td> <td>完全正确</td> </tr> <tr> <td>测试数据5</td> <td>完全正确</td> </tr> </tbody> </table>	测试数据	评判结果	测试数据1	完全正确	测试数据2	完全正确	测试数据3	完全正确	测试数据4	完全正确	测试数据5	完全正确
测试数据	评判结果													
测试数据1	完全正确													
测试数据2	完全正确													
测试数据3	完全正确													
测试数据4	完全正确													
测试数据5	完全正确													
			<a href="#">详细</a>											

上图中有10个线段，其中5、10、4号线段连成了一条连续线段，线段数3条，起点位置为5号线段的左端点；6、8、2、3、9号线段连成了另一条连续线段，该连续线段包含的线段数最多，为5条，起点位置是6号线段的左端点。

**注意：**

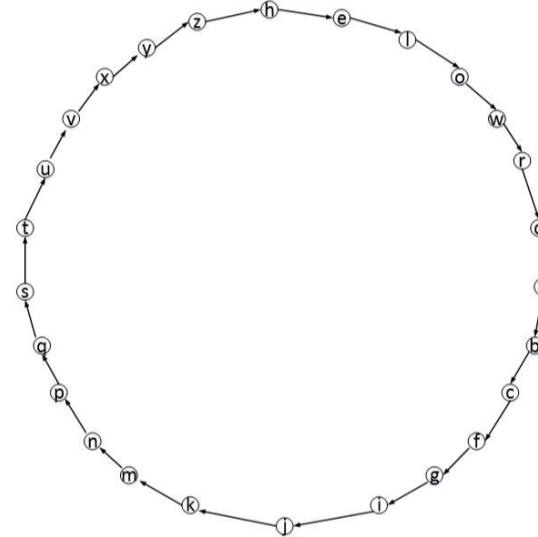
#	题目	分值	批阅信息
	<p>1) 不考虑线段中间相交的情况；      2) 不会有三条或三条以上的线段共一个端点；      3) 只会出现一条包含最多线段的连续线段；</p> <p><b>【输入形式】</b>      先从控制台输入线段数，然后从下一行开始分行输入各线段两个端点的x、y坐标，其中第一个端点的X轴坐标小于第二个端点的X轴坐标，即<math>x_1 &lt; x_2</math>，x、y坐标都用整数表示，不会超过int的表示范围。各整数坐标值间以一个空格分隔。</p> <p><b>【输出形式】</b>      先在控制台输出包含最多线段数的连续线段的线段数，然后输出连续线段的起点的x、y坐标，输出数据都以一个空格分隔。</p> <p><b>【样例输入】</b></p> <pre>10 80 75 125 75 60 40 80 55 80 55 90 20 140 120 195 205 10 111 70 165 22 35 43 43 22 175 80 205 43 43 60 40 90 20 125 60 70 165 140 120</pre> <p><b>【样例输出】</b></p> <pre>5 22 35</pre> <p><b>【样例说明】</b>      输入了十个线段，第一个线段两个端点分别为(80,75)和(125,75)，其它线段类似，如上图所示，这些线段所构成的连续线段中包含最多线段数的连续线段的线段数为5，起点为(22,35)，所以输出：5 22 35。</p> <p><b>【评分标准】</b>      通过所有测试点将得满分。提交程序名为line.c。</p>		

#	题目	分值	批阅信息												
2.	<p><a href="#">猴子选大王</a></p> <p><b>【问题描述】 (建议用链表实现)</b></p> <p>要从n只猴子中选出一位大王。它们决定使用下面的方法： n只猴子围成一圈，从1到n顺序编号。从第q只猴子开始，从1到m报数，凡报到m的猴子退出竞选，下一次又从退出的那只猴子的下一只开始从1到m报数，直至剩下的最后一只为大王。请问最后哪只猴子被选为大王。</p> <p><b>【输入形式】</b></p> <p>控制台输入三个整数n, m, q, 各整数间以一个空格分隔。</p> <p><b>【输出形式】</b></p> <p>输出最后选为大王的猴子编号。</p> <p><b>【样例输入】</b></p> <p>7 4 3</p> <p><b>【样例输出】</b></p> <p>4</p> <p><b>【样例说明】</b></p> <p>输入有7只猴子，从第3只猴子开始，从1到4报数。最后编号为4的猴子被选为大王。</p> <p><b>【评分标准】</b></p> <p>本题要求输出最后被选为大王的猴子编号，完全正确得20分，每个测试点4分。上传C语言文件名为monkey.c。</p>	20.00	<p><a href="#">下载源文件</a></p> <p>得分20.00 最后一次提交时间:2022-04-07 16:45:44 成功编译,但有警告信息. list.c: In function 'josephu': list.c:34:11: warning: 'q' may be used uninitialized in this function [-Wmaybe-uninitialized] q-&gt;link=p-&gt;link; ^ list.c:26:9: warning: 'p' may be used uninitialized in this function [-Wmaybe-uninitialized] p-&gt;link=list; ^ 共有测试数据:5 平均占用内存:1.227K 平均CPU时间:0.00577S 平均墙钟时间:0.00574S</p> <table border="1"> <thead> <tr> <th>测试数据</th> <th>评判结果</th> </tr> </thead> <tbody> <tr> <td>测试数据1</td> <td>完全正确</td> </tr> <tr> <td>测试数据2</td> <td>完全正确</td> </tr> <tr> <td>测试数据3</td> <td>完全正确</td> </tr> <tr> <td>测试数据4</td> <td>完全正确</td> </tr> <tr> <td>测试数据5</td> <td>完全正确</td> </tr> </tbody> </table>	测试数据	评判结果	测试数据1	完全正确	测试数据2	完全正确	测试数据3	完全正确	测试数据4	完全正确	测试数据5	完全正确
测试数据	评判结果														
测试数据1	完全正确														
测试数据2	完全正确														
测试数据3	完全正确														
测试数据4	完全正确														
测试数据5	完全正确														

[详细](#)

#	题目	分值	批阅信息												
3.	<a href="#">多项式相乘</a>	20.00	<a href="#">下载源文件</a>												
	【问题描述】 (建议用链表实现)		得分20.00 最后一次提交时间:2022-03-24 21:21:18												
	编写一个程序实现两个一元多项式相乘。		共有测试数据:5 平均占用内存:1.226K 平均CPU时间:0.00497S 平均墙钟时间:0.00496S												
	【输入形式】		<table border="1"> <thead> <tr> <th>测试数据</th><th>评判结果</th></tr> </thead> <tbody> <tr> <td>测试数据1</td><td>完全正确</td></tr> <tr> <td>测试数据2</td><td>完全正确</td></tr> <tr> <td>测试数据3</td><td>完全正确</td></tr> <tr> <td>测试数据4</td><td>完全正确</td></tr> <tr> <td>测试数据5</td><td>完全正确</td></tr> </tbody> </table>	测试数据	评判结果	测试数据1	完全正确	测试数据2	完全正确	测试数据3	完全正确	测试数据4	完全正确	测试数据5	完全正确
测试数据	评判结果														
测试数据1	完全正确														
测试数据2	完全正确														
测试数据3	完全正确														
测试数据4	完全正确														
测试数据5	完全正确														
	首先输入第一个多项式中系数不为0的项的系数和指数，以一个空格分隔。且该多项式中各项的指数均为0或正整数，系数和最高幂次不会超过int类型的表示范围。对于多项式 $a_nx^n + a_{n-1}x^{n-1} + \dots + a_1x^1 + a_0x^0$ 的输入方法如下：  $a_n \ n \ a_{n-1} \ n-1 \dots \ a_1 \ 1 \ a_0 \ 0$ 即相邻两个整数分别表示表达式中一项的系数和指数。在输入中只出现系数不为0的项。最后一项的指数后没有空格，只有一个回车换行符。 按照上述方式再输入第二个多项式。														
	【输出形式】		<a href="#">详细</a>												
	将运算结果输出到屏幕。将系数不为0的项按指数从高到低的顺序输出，每次输出其系数和指数，均以一个空格分隔，最后一项的指数后也可以有一个空格。														
	【样例输入】														
	10 80000 2 6000 7 300 5 10 18 0 3 6000 5 20 8 10 6 0														
	【样例输出】														
	30 86000 50 80020 80 80010 60 80000 6 1200 0 21 6300 10 6020 31 6010 66 6000 35 320 56 310 42 300 25 30 130 20 174 10 108 0														
	【样例说明】														
	输入的两行分别代表如下表达式： $10x^{80000} + 2x^{6000} + 7x^{300} + 5x^{10} + 18$ $3x^{6000} + 5x^{20} + 8x^{10} + 6$ 相乘结果为： $30x^{86000} + 50x^{80020} + 80x^{80010} + 60x^{80000} +$ $6x^{12000} + 21x^{6300} + 10x^{6020} + 31x^{6010} + 66x^{6000} + 35x^{320} + 56x^{310} + 42x^{300} + 25x^{30} + 13$ $0x^{20} + 174x^{10} + 108$														
	提示：利用 <a href="#">链表</a> 存储多项式的系数和指数。														
	【评分标准】														
	该题要求输出相乘后多项式中系数不为0的系数和指数，共有5个测试点。上传C语言文件名为multi.c。														
4.	<a href="#">文件加密(环)</a>	20.00	<a href="#">下载源文件</a>												
	【问题描述】 (建议用链表实现)		得分20.00 最后一次提交时间:2022-03-26 13:29:19												
	有一种文本文件加密方法，其方法如下：		成功编译,但有警告信息。 encode.c: In function 'main': encode.c:12:2: warning: implicit declaration of function 'gets' [-Wimplicit-function-declaration] gets(s); ^ encode.c:16:3: warning: array subscript has type 'char' [-Wchar-subscripts] if(!v[s[i]]) ^ encode.c:18:4: warning: array subscript has type 'char' [-Wchar-subscripts] v[s[i]]=1;												
	1、密钥由所有ASCII码可见字符 (ASCII码编码值32-126为可见字符) 组成，密钥长度不超过32个字符；														
	2、先将密钥中的重复字符去掉，即： <b>只保留最先出现的字符</b> ，其后出现的相同字符都去掉；														

#	题目	分值
3、将不含重复字符的密钥和其它不在密钥中的可见字符（按字符升序）连成一个由可见字符组成的环，密钥在前，密钥的头字符为环的起始位置；		批阅信息 encode.c:36:3: warning: array subscript has type 'char' [-Wchar-subscripts] to[ss[p]]=ss[q]; ^ encode.c:39:2: warning: array subscript has type 'char' [-Wchar-subscripts] to[ss[p]]=ss[0]; ^ encode.c:47:4: warning: array subscript has type 'char' [-Wchar-subscripts] if(opt[i]>=l&&opt[i]<=r) fprintf(out,"%c",to[opt[i]]); ^ /tmp/ccwp9cJr.o: In function `main': encode.c:(.text.startup+0x30): warning: the `gets' function is dangerous and should not be used. 共有测试数据:5 平均占用内存:1.395K 平均CPU时间:0.00694S 平均墙钟时间:0.00691S
4、设原密钥的第一个字符（即环的起始位置）作为环的开始位置标识，先从环中删除第一个字符（位置标识则移至下一个字符），再沿着环从下一个字符开始顺时针以第一个字符的ASCII码值移动该位置标识至某个字符，则该字符成为第一个字符的密文字符；然后从环中删除该字符，再从下一个字符开始顺时针以该字符的ASCII码值移动位置标识至某个字符，找到该字符的密文字符；依次按照同样方法找到其它字符的密文字符。当环中只剩一个字符时，则该剩下的最后一个字符的密文为原密钥的第一个字符。		测试数据 评判结果 测试数据1 完全正确 测试数据2 完全正确 测试数据3 完全正确 测试数据4 完全正确 测试数据5 完全正确
下面以可见字符集只由小写字母组成为例来说明对应密文字符集生成过程。如果密钥为：hel loworld，将密钥中重复字符去掉后为：helowr d，将不在密钥中的小写字母按照升序添加在密钥后，即形成字符串：helowrdabcfghijklmn pqstuvwxyz，该字符串形成的环如下图所示：		详细 



明码的第一个字母为h, h也是环的起始位置。h的ASCII码制为104，先把h从环中删除，再从下一个字母e开始顺时针沿着环按其ASCII值移动位置标识（即：在字母e为移动第1次，共移动位置标识104次）至字母w，则h的密文字符为w。w的ASCII码制为119，然后将w从环中删除，再从下一个字母r开始顺时针沿着环移动位置标识119次至字母为l，则w的密文字符为l。依次按照同样方法找到其它字母的密文字符。环中剩下的最后一个字母为x，则x的密文字符为明码的第一个字母h。按照这种方法形成的密文转换字符表为：

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
q	g	k	c	d	r	o	w	x	u	t	s	f	i	m	b	z	a	y	v	n	j	l	h	p	e

上方为原文字符，下方为对应的密文字符。由所有ASCII可见字符集组成的字符集密文字符生成方式与上例相同。

编写程序实现上述文件加密方法。密钥从标准输入读取，待加密文件为当前目录下的in.txt文件，该文件中的字符若是可见字符，则按照上述方法进行加密，否则原样输出（例如：回车换行符），加密后生成的密文文件为当前目录下的in\_crpyt.txt。

【输入形式】

#	题目	分值	批阅信息
	密钥是从标准输入读取的一行字符串，可以包含任意ASCII码可见字符（ASCII码编码值32-126为可见字符），长度不超过32个字符。		
	【输出形式】		
	加密后生成的密文文件为当前目录下的in_crpyt.txt。		
	【样例输入】		
	C Programming(Second Edition)		
	假设in.txt文件内容为：		
	This book is meant to help the reader learn how to program in C. It is the definitive reference guide, now in a second edition. Although the first edition was written in 1978, it continues to be a worldwide best-seller. This second edition brings the classic original up to date to include the ANSI standard.		
	From the Preface:		
	【样例输出】		
	in_crpyt.txt文件内容为：		
	KgkIW#33>WkIW\^M8W83Wg\Z,W8g\WP \^u\PWZ\^PMWg3jW83W,P30P^AWkMWX5 W.8WkIW8g\Wu\EkMk8kt\WP\E\P\MR\W0-ku \+WM3jWkMW^W\ R3MuW\uk8k3M5WIZ8g 3-0gW8g\WEkPl8W\uk8k3MWj^IWjPk88\MW kMW'71G+Wk8WR3M8kM-\ W83W#\W^Wj3 PZujku\W#\ 8J\ZZ\ P5WKgkIW\ R3MuW\uk8k 3MW#PkM0\W8g\WRZ^llkRW3Pk0kM^ZW-, W83Wu^8\W83WkMRZ-u\W8g\WIOY.WI8^ Mu^Pu5  4P3AW8g\WdP\E^R\( 【样例说明】		
	输入的密钥为C Programming(Second Edition)，由该密钥生成的字符串环中字符依次为：		
	C Progamin(SecdEt)!#\$%&'*+,-./012345678 9:<=>?@ABDFGHJKLNOQRTUVWXYZ[\]^_`bfhjklpqsvuvxyz{}~		
	形成的字符转换字符表（第一行为原ASCII字符，第二行为对应的密文字符）为：		
			
	按照该密文转换字符表将in.txt中的内容转换成加密字符输出到in_crpyt.txt中。		
	【评分标准】		
	该题要求对指定的文件进行加密，提交的文件名为：encode.c。		
5.	<a href="#">词频统计（数组或链表实现）</a>	20.00	<a href="#">下载源文件</a>
	【问题描述】		得分20.00 最后一次提交时间:2022-03-25 22:17:28
	编写程序统计一个英文文本文件中每个单词的出现次数（词频统计），并将统计结果按单词字典序输出到屏幕上。		共有测试数据:5 平均占用内存:1.227K 平均CPU时间:0.00522S 平均墙钟时间:0.00520S

#	题目	分值	测试数据 批阅信息	评判结果
	注：在此单词为仅由字母组成的字符串序列。包含大写字母的单词应将大写字母转换为小写字母后统计。		测试数据1	完全正确
	【输入形式】		测试数据2	完全正确
	打开当前目录下文件“article.txt”，从中读取英文单词进行词频统计。		测试数据3	完全正确
	【输出形式】		测试数据4	完全正确
	程序将单词统计结果按单词字典序输出到屏幕上，每行输出一个单词及其出现次数，单词和其出现次数间由一个空格分隔，出现次数后无空格，直接为回车。		测试数据5	完全正确

[详细](#)**【样例输入】**

当前目录下文件article.txt内容如下：

“Do not take to heart every thing you hear.”

“Do not spend all that you have.”

“Do not sleep as long as you want;”

**【样例输出】**

all 1

as 2

do 3

every 1

have 1

hear 1

heart 1

long 1

not 3

sleep 1

spend 1

take 1

that 1

thing 1

to 1

want 1

you 3

**【样例说明】**

按单词字典序依次输出单词及其出现次数。

**【评分标准】**

通过所有测试点将得满分。

北京航空航天大学

若重置密码，请与当前的任课教师联系