

清华大学本科生考试试题专用纸

考试课程 计算机语言与程序设计 (卷) 2011 年 1 月 12 日

姓名 _____ 班级 _____ 学号 _____

大题	题号	一	二	三	四	总分
	分数	30 分	25 分	25 分	20 分	
成绩						

试卷提交说明

1. 在考卷上填写你的姓名、学号；
2. 按要求编程，调试运行，将每道题的结果考屏，粘贴在考卷对应的题后；
3. 交卷时，提交一个压缩文件包，包括：
 - a) 文件包名
学号+姓名，如：2011123456 张三.rar
 - b) 文件包内容
 - ✧ 每道题的 c 工程文件夹；
 - ✧ 考卷。

一、字符串处理。编制一个程序，要求实现如下功能：

1. 主函数 (10 分)
按程序运行示例 1 编程，当且仅当输入 '@' 时，程序结束运行。
2. 函数 input() (10 分)
从键盘输入字符串 s (长度 < 40 个字符，无空格)、以及整型数 i，并传回给主函数；
3. 函数 split() (10 分)
 - a) split() 接受从主函数传递过来的、input() 输入的字符串 s 和 i ($0 < i < n-1$)，并将 s 拆分成两个新的字符串 s1 和 s2，拆分点是 s+i；
 - b) 拆分后，s1 由 s[0] ~ s[i-1] 元素序列组成，s2 则包含了 s 的剩余元素序列 s[i], s[i+1], ...；
 - c) split() 将 s1、s2 传回给主函数输出至屏幕。

程序运行示例 1

input 输入字符串 a 是 "123456789"、i=7，则拆分后新的字符串 s1、s2 分别是：

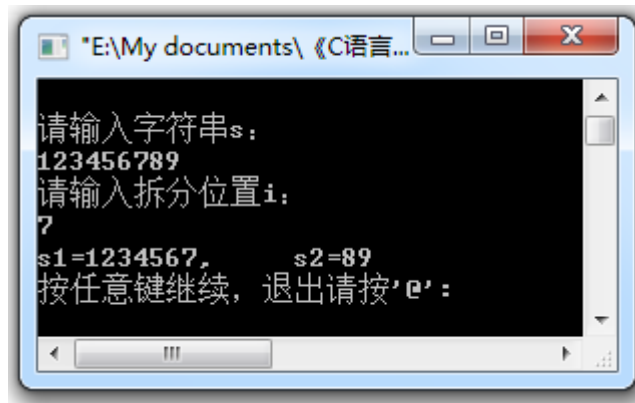
```
#include<stdio.h>
#include<malloc.h>
int input(char *a)
{
    int m,n=0,w;
    char i,ch;
    fflush(stdin);
    printf("please enter a number:\n");
    for(i=0;i<40;i++)
    {
        ch=getchar();
        if(ch=='\n')break;
        a[i]=ch;
        n++;
    }
    a[n]='\0';
    fflush(stdin);
    printf("please enter a number for i:\n");
    scanf("%d",&w);
    return(w);
}
```

```

void split(char *a,char *b,char *c,int u)
{
    int i,j;
    for(i=0;i<u;i++)
    {
        b[i]=a[i];
    }
    b[i]='\0';
    for(j=u;a[j]!='\0';j++)
    {
        c[j-u]=a[j];
    }
    c[j-u]='\0';
}

void main()
{
    char v[40],s1[40],s2[40];
    int i,j,k,h;
    h=input(v);
    //printf("%s %d\n",v,h);
    split(v,s1,s2,h);
    printf("s1=%s,      s2= %s\n",s1,s2);
}

```



```

#include<stdio.h>
#include<malloc.h>
#include<string.h>
void main()
{
    void input(int *i,char *s);
    void split(int i,char *s,char *s1,char *s2);
    int i,*a;
    char s[40],*s1,*s2,ch;
    a=&i;
    while(ch!='@')
    {
        input(a,s);
        puts(s);
        s2=(char*)malloc(strlen(s)-i+1);
        s1=(char*)malloc(i+1);
        split(i,s,s1,s2);
        puts(s1);
        puts(s2);
    }
}

```

```

        printf("按任意键继续，若使程序结束运行，输入'@'");
        ch=getchar();
    }
}
void input(int *i,char *s)
{
    printf("i:=\n");
    scanf("%d",i);
    printf("请输入 s:\n");
    scanf("%s",s);
    fflush(stdin);
}
void split(int i,char *s,char *s1,char *s2)
{
    int j;
    for(j=0;j<i;j++)
    {
        *(s1+j)=*(s+j);
    }
    *(s1+j)='\0';
    for(;*(s+j)!='\0';j++)
    {
        *(s2+j-i-1)=*(s+j);
    }
    *(s2+j)='\0';
}

```

二、括弧匹配。用字符串（长度最大 80）模拟一条 c 语句，如：

```
if((a<b)&&(a<c))min=a;
```

编程（**不允许调用任何 c 或 c++ 库函数，输入输出操作除外**）：

1. 主函数（5 分）

按**程序运行示例 2** 编程，当且仅当输入 '@' 时，程序结束运行。

2. Input 函数（5 分）

读入字符串，传回给主函数；

3. Matching 函数

从左至右扫描字符串，判别该字符串内的左、右圆括弧是否平衡：

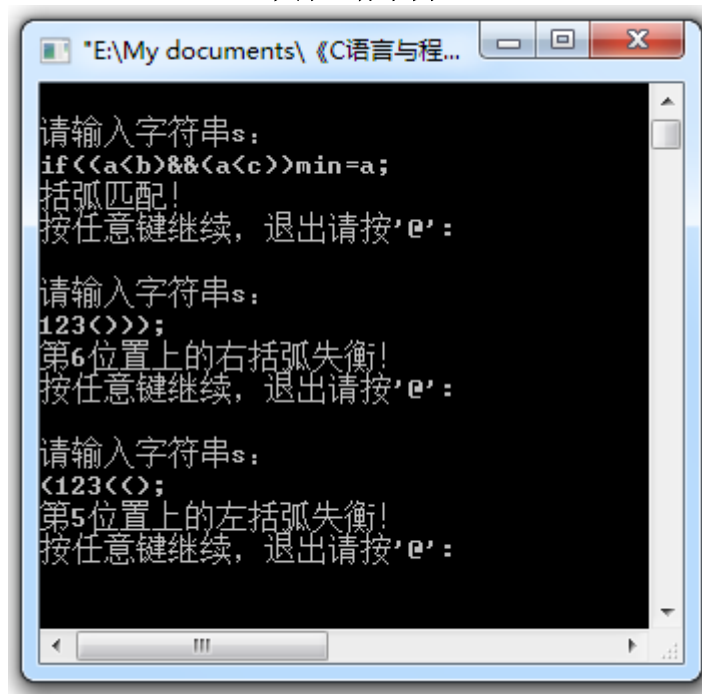
a) 若平衡，则返回正常匹配信息；（5 分）

b) 如果字符串不平衡，返回字符串中、第一个不匹配的圆括弧位置，即：

- **右圆括弧**：检索到第一个没有左圆括弧与之相配的右圆括弧时，中断扫描，返回该右圆括弧在字符串中的位置；（5 分）

- **左圆括弧**：检索到第一个没有右圆括弧与之相配的左圆括弧时，中断扫描，返回该左圆括弧在字符串中的位置。（5 分）

程序运行示例 2



```
#include<stdio.h>
#include<malloc.h>
#include<string.h>
void main()
{
    void input(char *s);
    int Matching(char *s);
    char s[40],ch;
    int d;
    while(ch!='@')
    {
        input(s);
        d=Matching(s);
```

```

        if(d!=0)
        {
            printf("The wrong position is NO.%.d.\n",d);
        }
        else printf("Correct!\n");
printf("按任意键继续，若使程序结束运行，输入'@'");
        ch=getchar();
    }
}

void input(char *s)
{
    printf("请输入 s:\n");
    fflush(stdin);
    gets(s);
    fflush(stdin);
}

int Matching(char *s)
{
    int i,j=0,n=0;
    int *p;
    p=(int*)malloc(strlen(s)*sizeof(int));
    *p=-1;
    for(i=0;*(s+i)!='\0';i++)
    {
        if(*(s+i)=='(')
        {
            if(j==0)
            {
                *(p+j)=i;
                j++;
            }
            else if(*(s+*(p+j-1))=='(')
            {
                *(p+j-1)=-1;
                j--;
            }
            else
            {
                j++;
                *(p+j-1)=i;
            }
        }
        else if(*(s+i)==')')
        {

```

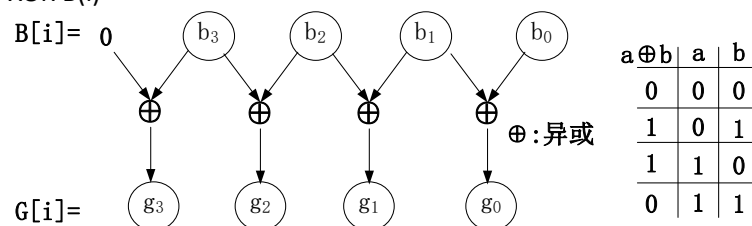
```
        *(p+j)=i;
        j++;
    }
}
if(*p== -1)return(0);
else return(*p+1);
}第 2 页/共 5 页
```

三、格雷码转换。格雷码(Gray code)又叫循环二进制码或反射二进制码，是一种错误最小化的二进制编码方式，如下是十进制、4 位自然二进制码与 4 位格雷码的对照表。

十进制数	自然二进制数	格雷码
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

➤ 二进制→格雷码转换快速方法（G：格雷码 B：二进位码）

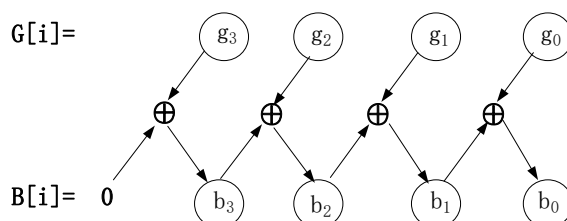
$$G(i) = B(i+1) \text{ XOR } B(i)$$



➤ 格雷码→二进位码

$$B(i) = B(i+1) \text{ XOR } G(i)$$

格雷码→二进制图解（4位）



编程：（不允许调用任何 c 或 c++ 库函数，输入输出操作除外）

1. 主函数（10 分）

按程序运行示例 3 编程；

2. 函数 input()（5 分）

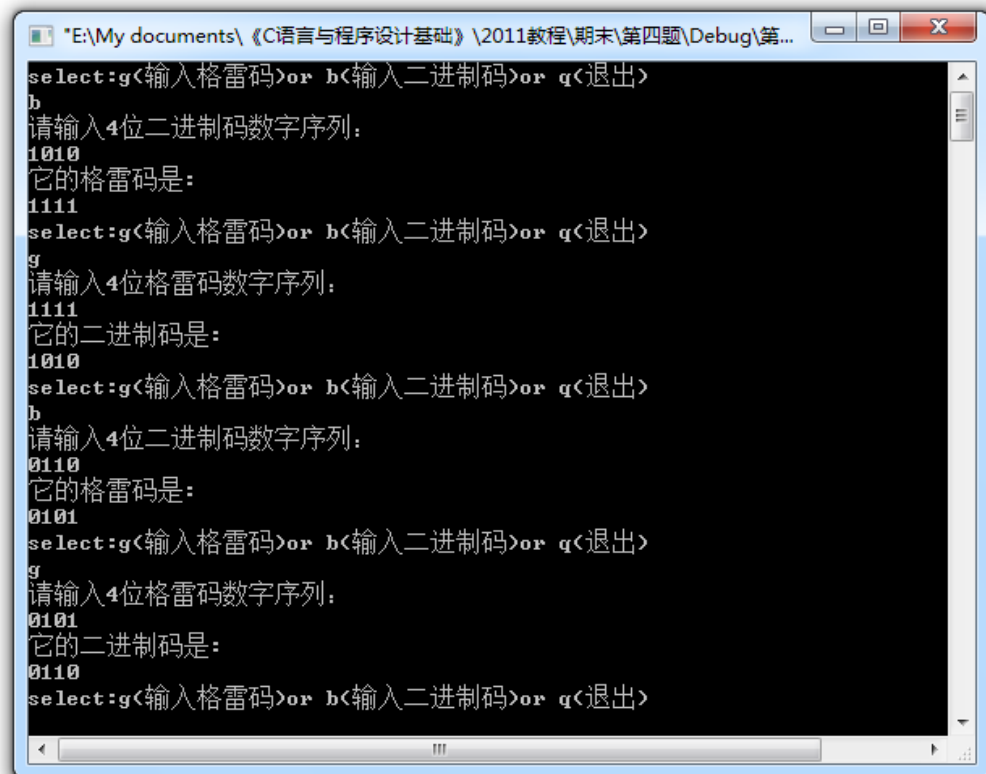
从键盘输入长度为 4 的字符串 s 并传回主函数，s 的元素由 ASCII 码的 0、1 数字组成，高位在前（字符串头部）、低位在后（字符串尾部）；

3. 函数 gray(char *s, int flag):

a) 若 flag=1，输入的 s 为二进制码，gray 将 s 转换成格雷码，传回主函数；（5 分）

b) 若 flag=-1，输入的 s 为格雷码，gray 将 s 转换成二进制码，传回主函数。（5 分）

程序运行示例 3



```
select:g<输入格雷码>or b<输入二进制码>or q<退出>
b
请输入4位二进制码数字序列:
1010
它的格雷码是:
1111
select:g<输入格雷码>or b<输入二进制码>or q<退出>
g
请输入4位格雷码数字序列:
1111
它的二进制码是:
1010
select:g<输入格雷码>or b<输入二进制码>or q<退出>
b
请输入4位二进制码数字序列:
0110
它的格雷码是:
0101
select:g<输入格雷码>or b<输入二进制码>or q<退出>
g
请输入4位格雷码数字序列:
0101
它的二进制码是:
0110
select:g<输入格雷码>or b<输入二进制码>or q<退出>
```

```
#include<stdio.h>
#include<malloc.h>
char *input1()
{
    char *a,ch;
    int i;
    a=(char*)malloc(5*sizeof(char*));
    printf("请输入 4 位格雷码数字序列: \n");
    fflush(stdin);
    for(i=0;i<4;i++)
    {
        ch=getchar();
        if(ch=='\n')break;
        a[i]=ch;
    }
    //a[i]='\0';
    return(a);
}
char *input2()
{
    char *a,ch;
    int i;
    a=(char*)malloc(5*sizeof(char*));
    printf("请输入 4 位二进制数字序列: \n");
    fflush(stdin);
    for(i=0;i<4;i++)
    {
        ch=getchar();
        if(ch=='\n')break;
```



```

    a[i]=ch;
}
//a[i]='\0';
return(a);
}
void gray(char *s,int flag)
{
    if(flag==1)
    {
        if(s[0]=='0'&& s[1]=='0'&& s[2]=='1'&& s[3]=='0')
        {
            s[0]='0';
            s[1]='0';
            s[2]='1';
            s[3]='1';
            //s[4]='\0';
        }
        else if(s[0]=='0'&& s[1]=='0'&& s[2]=='1'&& s[3]=='1')
        {
            s[0]='0';
            s[1]='0';
            s[2]='1';
            s[3]='0';
        }
        else if(s[0]=='0'&& s[1]=='1'&& s[2]=='0'&& s[3]=='0')
        {
            s[0]='0';
            s[1]='1';
            s[2]='1';
            s[3]='0';
        }
        else if(s[0]=='0'&& s[1]=='1'&& s[2]=='0'&& s[3]=='1')
        {
            s[0]='0';
            s[1]='1';
            s[2]='1';
            s[3]='1';
        }
        else if(s[0]=='0'&& s[1]=='1'&& s[2]=='1'&& s[3]=='0')
        {
            s[0]='0';
            s[1]='1';
            s[2]='0';
            s[3]='1';
        }
        else if(s[0]=='0'&& s[1]=='1'&& s[2]=='1'&& s[3]=='1')
        {
            s[0]='0';
            s[1]='1';
            s[2]='0';
            s[3]='0';
        }
        else if(s[0]=='1'&& s[1]=='0'&& s[2]=='0'&& s[3]=='0')
        {

```

```

        s[0]='1';
        s[1]='1';
        s[2]='0';
        s[3]='0';
    }
    else if(s[0]=='1'&&s[1]=='0'&&s[2]=='0'&&s[3]=='1')
    {
        s[0]='1';
        s[1]='1';
        s[2]='0';
        s[3]='1';
    }
    else if(s[0]=='1'&&s[1]=='0'&&s[2]=='1'&&s[3]=='0')
    {
        s[0]='1';
        s[1]='1';
        s[2]='1';
        s[3]='1';
    }
    else if(s[0]=='1'&&s[1]=='0'&&s[2]=='1'&&s[3]=='1')
    {
        s[0]='1';
        s[1]='1';
        s[2]='1';
        s[3]='0';
    }
    else if(s[0]=='1'&&s[1]=='1'&&s[2]=='0'&&s[3]=='0')
    {
        s[0]='1';
        s[1]='0';
        s[2]='1';
        s[3]='0';
    }
    else if(s[0]=='1'&&s[1]=='1'&&s[2]=='0'&&s[3]=='1')
    {
        s[0]='1';
        s[1]='0';
        s[2]='1';
        s[3]='1';
    }
    else if(s[0]=='1'&&s[1]=='1'&&s[2]=='1'&&s[3]=='0')
    {
        s[0]='1';
        s[1]='0';
        s[2]='0';
        s[3]='1';
    }
    else if(s[0]=='1'&&s[1]=='1'&&s[2]=='1'&&s[3]=='1')
    {
        s[0]='1';
        s[1]='0';
        s[2]='0';
        s[3]='0';
    }
}

```

```

}
else if(flag== -1)
{
    if(s[0]=='0'&& s[1]=='0'&& s[2]=='1'&& s[3]=='1')
    {
        s[0]='0';
        s[1]='0';
        s[2]='1';
        s[3]='0';
    }
    else if(s[0]=='0'&& s[1]=='0'&& s[2]=='1'&& s[3]=='0')
    {
        s[0]='0';
        s[1]='0';
        s[2]='1';
        s[3]='1';
    }
    else if(s[0]=='0'&& s[1]=='1'&& s[2]=='1'&& s[3]=='0')
    {
        s[0]='0';
        s[1]='1';
        s[2]='0';
        s[3]='0';
    }
    else if(s[0]=='0'&& s[1]=='1'&& s[2]=='1'&& s[3]=='1')
    {
        s[0]='0';
        s[1]='1';
        s[2]='0';
        s[3]='1';
    }
    else if(s[0]=='0'&& s[1]=='1'&& s[2]=='0'&& s[3]=='0')
    {
        s[0]='0';
        s[1]='1';
        s[2]='1';
        s[3]='1';
    }
    else if(s[0]=='1'&& s[1]=='1'&& s[2]=='0'&& s[3]=='0')
    {
        s[0]='1';
        s[1]='0';
        s[2]='0';
        s[3]='0';
    }
    else if(s[0]=='1'&& s[1]=='1'&& s[2]=='0'&& s[3]=='1')
    {
        s[0]='1';
        s[1]='0';
        s[2]='0';
        s[3]='1';
    }
    else if(s[0]=='1'&& s[1]=='1'&& s[2]=='1'&& s[3]=='1')
    {

```

```

        s[0]='1';
        s[1]='0';
        s[2]='1';
        s[3]='0';
    }
    else if(s[0]=='1'&& s[1]=='1'&& s[2]=='1'&& s[3]=='0')
    {
        s[0]='1';
        s[1]='0';
        s[2]='1';
        s[3]='1';
    }
    else if(s[0]=='1'&& s[1]=='0'&& s[2]=='1'&& s[3]=='0')
    {
        s[0]='1';
        s[1]='1';
        s[2]='0';
        s[3]='0';
    }
    else if(s[0]=='1'&& s[1]=='0'&& s[2]=='1'&& s[3]=='1')
    {
        s[0]='1';
        s[1]='1';
        s[2]='0';
        s[3]='1';
    }
    else if(s[0]=='1'&& s[1]=='0'&& s[2]=='0'&& s[3]=='1')
    {
        s[0]='1';
        s[1]='1';
        s[2]='1';
        s[3]='0';
    }
    else if(s[0]=='1'&& s[1]=='0'&& s[2]=='0'&& s[3]=='0')
    {
        s[0]='1';
        s[1]='1';
        s[2]='1';
        s[3]='1';
    }
}
}
void main()
{
    int z=0;
    char n,*b;
    int a,i;
    while(z==0)
    {
        printf("select:g(输入格雷码)or b(输入二进制码)or q(退出)\n");
        scanf("%c",&n);
        //printf("n is %d",n);
        if(n==103)
        {

```

```

        a=-1;
        b=input1();
        fflush(stdin);
        gray(b,a);
        printf("它的二进制码是: ");
        for(i=0;i<4;i++)printf("%c",b[i]);
        printf("\n");
    }
    else if(n==98)
    {
        a=1;
        b=input2();
        fflush(stdin);
        gray(b,a);
        printf("它的格雷码是: ");
        for(i=0;i<4;i++)printf("%c",b[i]);
        printf("\n");
    }
    else if(n==113) break;
    else printf("请输入正确的指令\n");
    fflush(stdin);
}
} #include<stdio.h>
#include<malloc.h>
#include<string.h>
void main()
{
    void input(char *s);
    char*gray(char *s, int flag);
    char *s,ch;
    s=(char*)malloc(5);
    while(1)
    {
        printf("g:格雷码,b:二进制,q:结束");
        ch=getchar();
        input(s);
        if(ch=='g')
            s=gray(s,-1);
        if(ch=='b')
            s=gray(s,1);
        if(ch=='q')break;
        puts(s);
    }
}
void input(char *s)
{
    fflush(stdin);
    printf("请输入: ");
    gets(s);
}
char*gray(char *s, int flag)
{
    char *s1;
    int i;

```

```

s1=(char*)malloc(5);
if(flag==-1)
{
    if(*(s)=='1')*(s1)='1';
    else *(s1)='0';
    for(i=1;i<4;i++)
    {
        if(*(s+i)!='*(s+i-1))*(s1+i)='1';
        else *(s1+i)='0';
    }
}
if(flag==1)
{
    if(*(s)=='1')
    {
        *(s1)='1';
    }
    else
    {
        *(s1)='0';
    }
    for(i=1;i<4;i++)
    {
        *(s1+i)=(*(s1+i-1)-48)^(*(s+i)-48)+48;
    }
}
*(s1+4)='\0';
free(s);
return(s1);
}

```

四、二维矩阵存储。对任意一个 $N \times M$ 的整数型矩阵 a ($N > 3, M > 4$)，编程（不允许调用任何 **c 或 c++ 库函数，输入操作除外**）实现如下功能：

1. 函数 **input()** （8 分）

从键盘输入矩阵 a ，传回给主函数；

2. 函数 **RowInsert()** （5 分）

a) RowInsert()将 a 的第 i 行存储成一个循环链表，节点结构如下：

```
struct node{
    int aij;
    struct node *next;
};
```

b) 将行链的头指针返回给主调函数；

3. 函数 **Link Layer ()** （4 分）

Link Layer ()逐行调用 RowInsert()，将矩阵 a 存储成链表，并将链式存储结构的 a 传回给主函数；

4. 函数 **list()** （3 分）

主函数调用 list()，遍历输出链式存储结构的矩阵 a ，格式如程序运行示例 4。

程序运行示例 4（矩阵 $a[4][5]$ 的链式存储与输出）

```
#include<stdio.h>
#include<malloc.h>
#define NULL 0
#define LEN sizeof(struct node)
struct node
{
    int num;
    struct node *next;
};
struct node *Row_insert(int *a)
{
    struct node *head,*p1,*p2;
    p1=p2=(struct node*)malloc(LEN);
    p1->num=a[0];
    head=p1;
    p2=p1;
    int i;
    for(i=1;i<5;i++)
    {
        p1=(struct node*)malloc(LEN);
        p1->num=a[i];
        p2->next=p1;
        p2=p1;
    }
    p2->next=NULL;
    return(head);
}
struct node *Link_layer(int a[][100])
{
    struct node *head,*p1,*p2;
    int z=0,i;
    head=Row_insert(a[0]);
    p1=head;
    for(i=1;i<4;i++)
    {
        while(p1!=NULL)
        {
```

```

        p2=p1;
        p1=p1->next;
    }
    p2->next=Row_insert(a[i]);
    p1=p2;
}
return(head);
}
void print(struct node *head)//
{
    struct node *p;
    int z=0;
    //printf("These records are:\n");
    p=head;
    if(head!=NULL)
        do
        {
            printf("%d->",p->num);
            p=p->next;
            z++;
            if(z==5)
            {
                printf("\b\b  \n");
                z=0;
            }
        }
        while(p!=NULL);
    //printf("\b \n");
    //printf("%d %d\n",p->num,p->sala);
}
void main()
{
    struct node *b;
    int a[100][100];
    int m=4,n=5,i,j,k;
    //printf("please enter the number for m and n: ");
    //scanf("%d%d",&m,&n);
    for(i=0;i<m;i++)
    {
        printf("请输入第%d 行:\n",i);
        for(j=0;j<n;j++)
        {
            scanf("%d",&a[i][j]);
        }
        a[i][j]='\0';
    }
    b=Link_layer(a);
    print(b);
    //for(i=0;i<m;i++)
    //{
    //    for(j=0;j<n;j++)
    //    {
    //        printf("%d ",a[i][j]);
    //    }
    //    printf("\n");
    //}
}

```



```

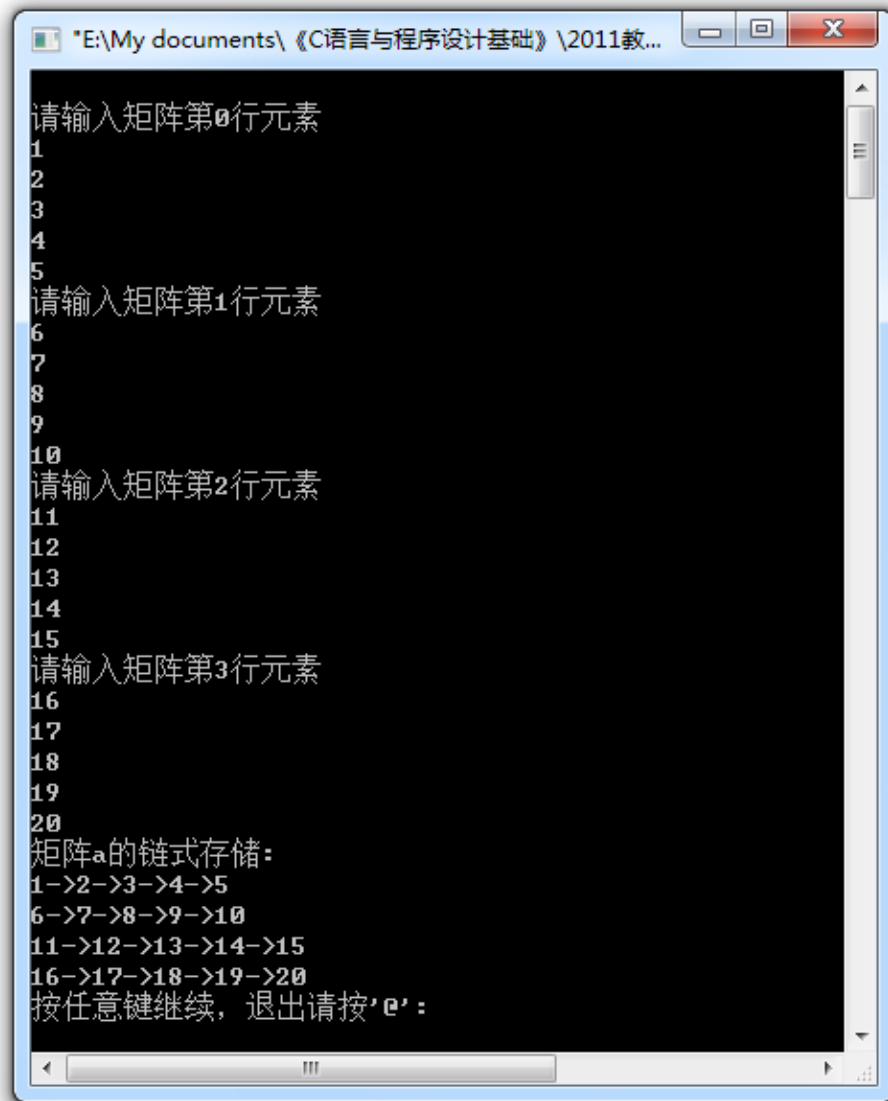
#include<stdio.h>
#include<malloc.h>
#include <time.h>
struct node
{
int aij;
struct node *next;
};
#define LEN sizeof(struct node)
struct node *RowInsert(int*s,int N)//建立链表//
{
int i;
struct node *head;
struct node *p1,*p2;
p1=p2=(struct node*)malloc(LEN);
for(i=0;i<N;i++)
{
if(i==0)
{
head=p1;
p1->aij=*(s+i);
}
else
{
p2->next=p1;
p2=p1;
p1=(struct node*)malloc(LEN);
p1->aij=*(s+i);
}
}
p2->next=p1;
p1->next=NULL;
return(head);
}
void list(struct node *head)//输出链表//
{
struct node *p;
int t=0;
p=head;
printf("打印单链表中全部节点: \n");
do{
printf("%4d",p->aij);
p=p->next;
t++;if(t%5==0)printf("\n");
}while(p!=NULL);
}
void main()
{
int **input(int m,int n);
struct node *RowInsert(int*s,int N);
struct node *LinkLayer(int**s,int M,int N);
void list(struct node *head);
int **a,m,n;
struct node *head;
printf("m,n=\n");
scanf("%d,%d",&m,&n);
a=input(m,n);

```

```

    head=LinkLayer(a,m,n);
    list(head);
}
struct node *LinkLayer(int**s,int M,int N)
{
    struct node *RowInsert(int*s,int N);
    int i;
    struct node *p1,*p2,*head;
    for(i=0;i<M;i++)
    {
        if(i==0)p1=head=RowInsert(*s,N);
        else
        {
            p2=p1;
            while(p2->next!=NULL)p2=p2->next;
            p1=RowInsert(*(s+i),N);
            p2->next=p1;
        }
    }
    return(head);
}
int **input(int m,int n) //建造一个 m 行,n 列二维数组//
{
    int **p;
    int i,j;
    p=(int**)malloc(m*sizeof(int*));
    for(i=0;i<m;i++)
    {
        *(p+i)=(int*)malloc(n*sizeof(int));
    }
    srand(time(NULL));
    for(i=0; i<m; i++)
    {
        for(j=0;j<n;j++)
        {
            (*(p+i)+j)=rand() %100;
            printf("%4d",*(p+i)+j));
        }
        printf("\n");
    }
    return(p);
}

```



The screenshot shows a Windows-style window titled "E:\My documents\《C语言与程序设计基础》\2011教...". The window contains a text area with the following text:

```
请输入矩阵第0行元素
1
2
3
4
5
请输入矩阵第1行元素
6
7
8
9
10
请输入矩阵第2行元素
11
12
13
14
15
请输入矩阵第3行元素
16
17
18
19
20
矩阵a的链式存储:
1->2->3->4->5
6->7->8->9->10
11->12->13->14->15
16->17->18->19->20
按任意键继续, 退出请按'q':
```

某一题:

```
#include<stdio.h>
#include<malloc.h>
int input(char *a)
{
    int m,n=0,w;
    char i,ch;
    fflush(stdin);
    printf("please enter a char:\n");
    for(i=0;i<40;i++)
    {
        ch=getchar();
        if(ch=='\n')break;
        a[i]=ch;
        n++;
    }
    //printf("this is %d\n",n);
    a[n]='\0';
    fflush(stdin);
    //printf("please enter a number for i:\n");
    //scanf("%d",&w);
```

```

return(n);
}
void input1(char *a,int m)
{
int n=0,z=0;
char i,j,ch;
fflush(stdin);
while(z==0)
{
    n=0;
    printf("please enter a number:\n");
    for(i=0;i<40;i++)
    {
        ch=getchar();
        if(ch=='\n')break;
        a[i]=ch;
        n++;
    }
    a[n]='\0';
    fflush(stdin);
    z=0;
    //printf("n is %d,m is %d",n,m);
    if(n==m)
    {
        int b=0;
        for(i=0;i<m;i++)
        {
            for(j=0;j<i;j++)
            {
                if(a[i]==a[j])
                {
                    b=1;
                    break;
                }
            }
            if(b==1)break;
            if(a[i]<48 || a[i]>48+m-1)
            {
                b=1;
                break;
            }
        }
        if(b==0)z=1;
    }
    if(z==1)break;
    else printf("ERROR!!");
}
//printf("please enter a number for i:\n");
//scanf("%d",&w);
//return(n);
}
void haha(char *a,char *b,char *c,int m)
{
    int i;

```

```
        for(i=0;i<m;i++)
        {
            c[b[i]-48]=a[i];
        }
        c[m]='\0';
    }
    void main()
    {
        char v1[80],v2[80],v3[80];
        int i,j,k,h;
        h=input(v1);
        input1(v2,h);
        //printf("%s %s\n",v1,v2);
        haha(v1,v2,v3,h);
        printf("%s\n",v3);
    }
```