

## 第四次作业：数组

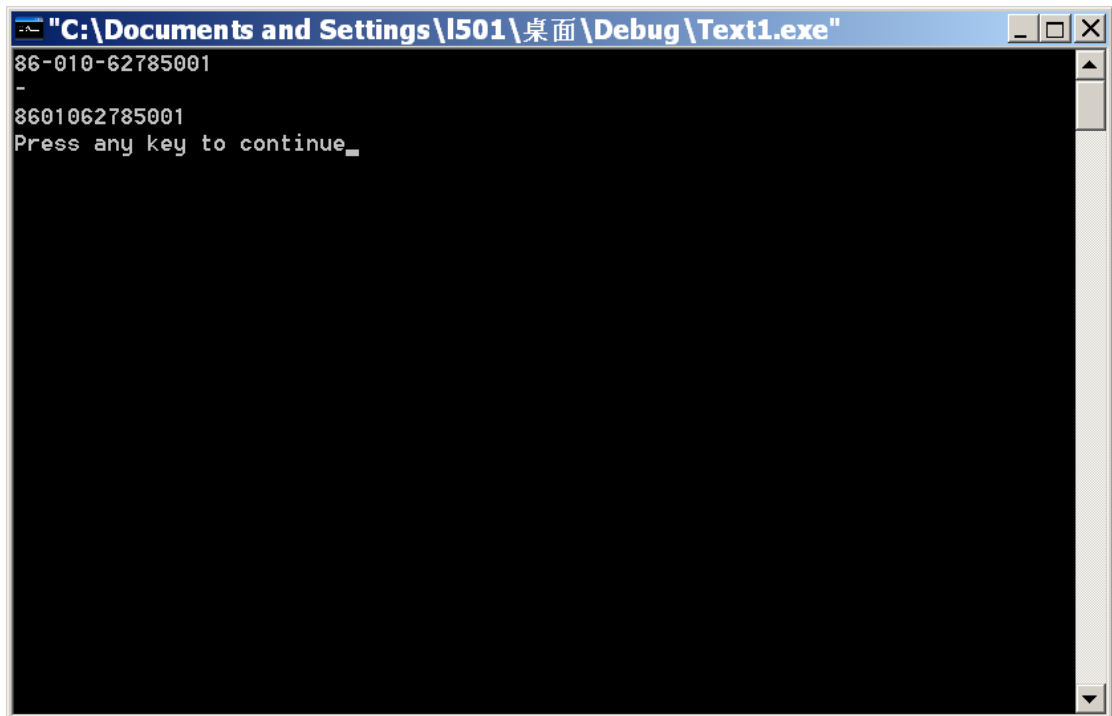
### 1. 字符筛选

编写一个程序，先输入一个字符串 str1（长度不超过 20），再输入一个单独的字符 ch，然后程序会把字符串 str 当中出现的所有的 ch 字符都删掉，从而得到一个新的字符串 str2，然后把这个字符串打印出来。例如：

- 输入：字符串 str1 为：86-010-62785001  
字符 ch 为：-
- 输出：字符串 str2 为：861062785001

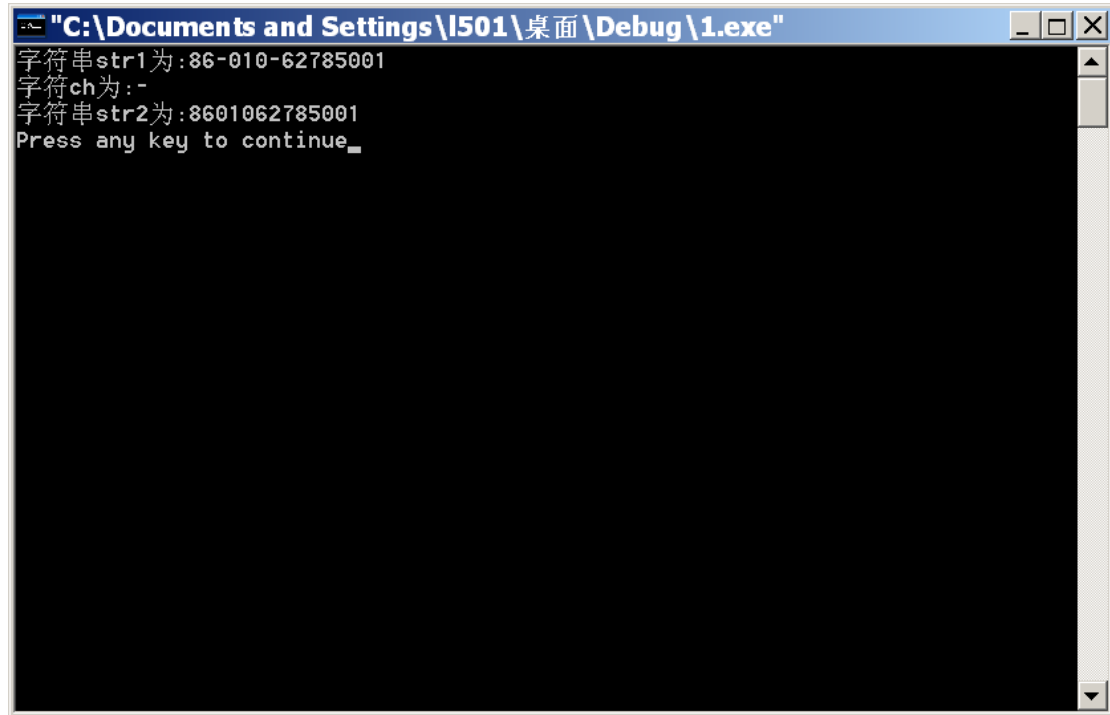
方法一：

```
#include<stdio.h>
void main()
{
    char str1[21],str2[21],ch;
    int i,j;
    printf("Input a string: ");
    gets(str1);
    printf("Input a character: ");
    ch=getchar();
    for(i=0,j=0;i<20;i++)
    {
        if(str1[i]!=ch)
            str2[j]=str1[i];
        else j++;
    }
    printf("%s\n", str2);
}
```



方法二：

```
#include <stdio.h>
#include <string.h>
void main()
{
    char ch,str1[20],str2[20];
    int i,j;
    gets(str1);
    puts(str1);
    scanf("%c",&ch);
    for((i=0,j=0);i<=strlen(str1);i++)
    {
        if(str1[i]!=ch)
        {
            str2[j]=str1[i];
            j++;
        }
    }
    printf("%s\n",str2);
}
```



## 2. 数据合并

编写一个程序，能够输入两个预先从小到大排列的整数型数组（假设数组长度最大为20，数组输入时遇到数字0结束）；要求将这两个数组的数字能够合并成为一个新的从小到大顺序排列的整数型数据，并将新数组输出。例如：

- 输入：请输入第一个数组：1, 2, 3, 12, 16, 20, 25, 40, 0  
请输入第二个数组：4, 5, 6, 15, 18, 20, 21, 22, 25, 38, 45, 0
- 输出：合并后的数组为：1, 2, 3, 4, 5, 6, 12, 15, 16, 18, 20, 20, 21, 22, 25, 25, 38, 40, 45

方法一：直接比较合并

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    int a[20],b[20],c[40],i,j,k,l;
```

```
    printf("请输入第一个数组： ");
```

```
    for(i=0;i<20;i++)
```

```
    {
```

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

```
        if (a[i]==0) break;
```

```
    }
```

```
    printf("请输入第二个数组： ");
```

```
    for(i=0;i<20;i++)
```

```
    {
```

```
        scanf("%d",&b[i]);
```

```
        if (b[i]==0) break;
```

```
    }
```

```

for(k=0,i=0,j=0;k<40;k++)
{
    if(a[i]!=0&&b[j]!=0)
    {
        if(a[i]<b[j])
        {
            c[k]=a[i];
            i++;
        }
        else
        {
            c[k]=b[j];
            j++;
        }
    }
    else if(a[i]!=0&&b[j]==0)
    {
        c[k]=a[i];
        i++;
    }
    else if(a[i]==0&&b[j]!=0)
    {
        c[k]=b[j];
        j++;
    }
    else
        break;
}
for(l=0;l<k;l++)
    printf("%d ",c[l]);
}

```

```

"D:\Software\Micorsoft\VC6.0\MyProjects\Test\Debug\Test.exe"
请输入第一个数组: 1 3 4 7 0
请输入第二个数组: 2 4 6 8 0
1 2 3 4 4 6 7 8 Press any key to continue_

```

方法二：先合并数组，再对数组排序

```
#include <stdio.h>

void main()
{
    int a[20], b[20], c[40], i, j, k;
    printf("输入第一个数组: ");
    for(i=0, k=0; i < 20; i++, k++)
    {
        scanf("%d", &a[i]);
        c[k] = a[i];
        if (a[i]==0) break;
    }
    printf("输入第二个数组: ");
    for(i=0; i<20; i++, k++)
    {
        scanf("%d", &b[i]);
        c[k] = b[i];
        if (b[i]==0) break;
    }
    for(j = 0; j < k-1; j++){
        for(i = 0; i < k-j-1; i++){
            if(c[i] > c[i+1]){
                int temp = c[i];
                c[i] = c[i+1];
                c[i+1] = temp;
            }
        }
    }
    for(i = 0; i < k; i++)
        printf("%d \n", c[i]);
}
```

### 3. 数制转换

请使用数组编写一个程序，输入一个二进制的字符串（长度不超过 32），然后计算出相应的十进制整数，并把它打印出来。例如，

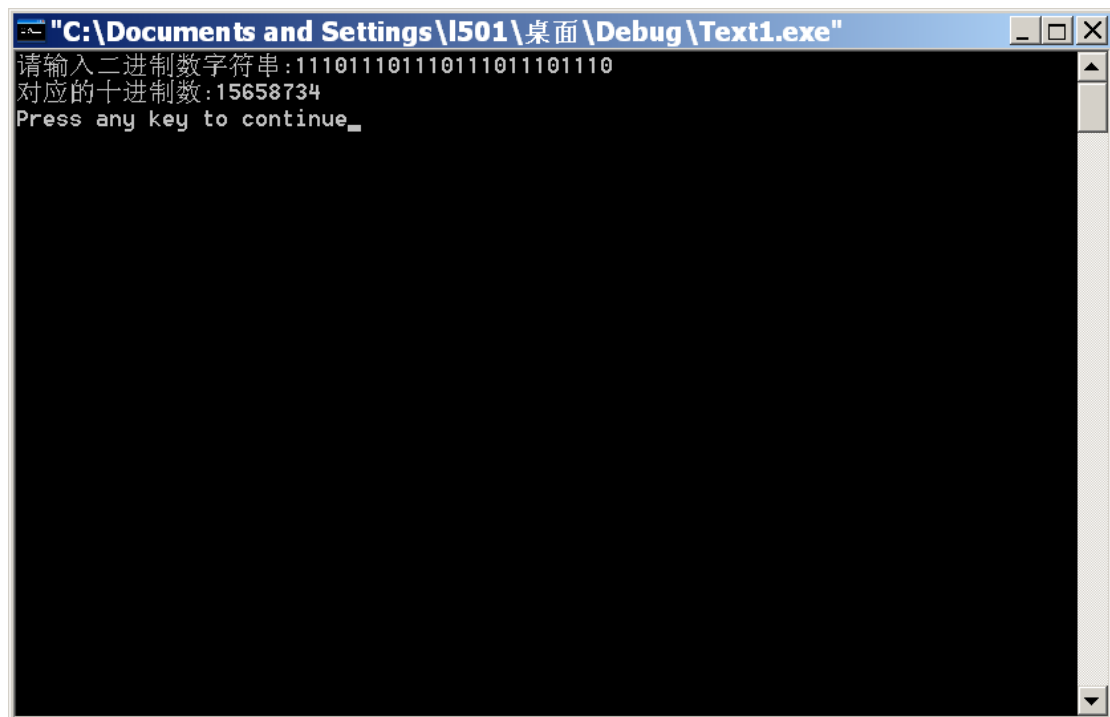
- 输入：请输入二进制数字字符串：111011101110111011101110
- 输出：对应的十进制数：15658734

```
#include<stdio.h>
#include<math.h>
void main()
{
    char c[32];
    int n,i,a[32],s=0;
```

```

printf("请输入二进制数字串:");
scanf("%s",c);
n=strlen(c)-1;
for (i=0;i<=n;i++)
{
    a[i]=c[i]-48;
    s=s+a[i]*pow(2,n-i);
}
printf("对应的十进制数:%d\n",s);
}

```



#### 4. 对称数字

一辆汽车在开始出发前其里程表上的读数是一个对称数 9 5 8 5 9，后匀速行驶两个小时后，发现里程表上是一个新的对称数。问该新的对称数是多少？汽车的速度是多少？

注：所谓对称数是指从左向右读与从右向左读完全一样的数字。

请在合理的范围内（提示，车速别太快也不能太慢，假设介于 30~120 公里/小时），编程解决上述问题。

```

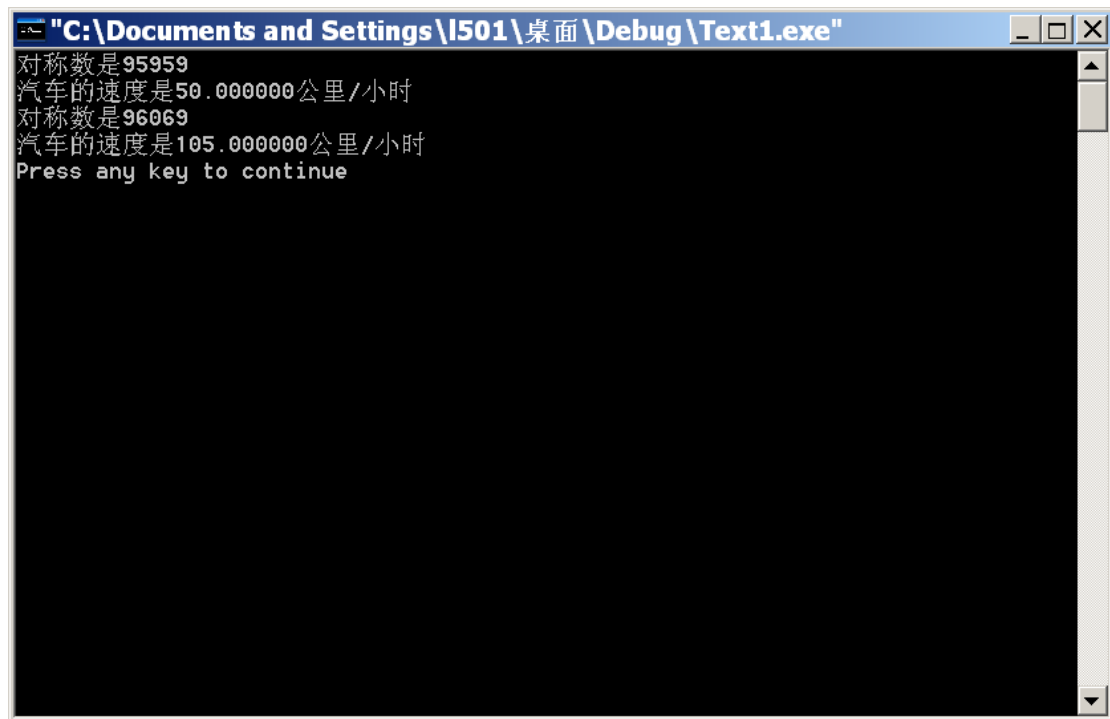
#include<stdio.h>
void main()
{
    int a[5],s1=95859,s2,s,i=0;
    for(s=60;s<=240;s++)
    {
        s2=s1+s;
        a[0]=s2/10000;
    }
}

```

```

a[1]=(s2-10000*a[0])/1000;
a[2]=(s2-10000*a[0]-1000*a[1])/100;
a[3]=(s2-10000*a[0]-1000*a[1]-100*a[2])/10;
a[4]=s2-10000*a[0]-1000*a[1]-100*a[2]-10*a[3];
if(a[0]==a[4]&&a[1]==a[3])
    printf("对称数是%d\n汽车的速度是%f公里/小时\n",s2,1.0*s/2);
}
}

```



## 5. 矩阵元素的运算

读入 1 个正整数  $n(1 \leq n \leq 10)$ , 再读入  $n \times n$  阶方阵  $a$ , 计算该矩阵除副对角线、最后一列和最后一行以外的所有元素之和。(副对角线为从矩阵的右上角至左下角的连线)

例: 括号内是说明, 可以不输入、不输出。

- 输入: 4 ( $n=4$ )

2 3 4 1

5 6 1 1

7 1 8 1

1 1 1 1

- 输出: sum=35

代码:

```
#include<stdio.h>
```

```
void main()
```

```
{
```

```
    int a[10][10],n,i,j,sum=0;
```

```

scanf("%d",&n);
for (i=0;i<=n-1;i++)
{
    for (j=0;j<=n-1;j++)
    {
        scanf("%d",&a[i][j]);
        if(i+j!=n-1&&i!=n-1&&j!=n-1)
            sum=sum+a[i][j];
    }
}
printf("sum=%d\n",sum);
}

```

```

"C:\Documents and Settings\I501\桌面\Debug\Text1.exe"
4
2 3 4 1
5 6 1 1
7 1 8 1
1 1 1 1
sum=35
Press any key to continue

```

## 6. 加法口诀表

要求使用二维数组求解下列问题。（提示：将加数、被加数、和放入一个二维数组中，再输出该数组）。输入 1 个正整数  $n$  ( $1 \leq n \leq 10$ )，输出一张 20 以内的加法口诀表。要求，该口诀表中的加数与被加数都不大于  $n$ ，分别存放在该表格中的第一行和第一列。样式例如下：括号内是说明

- 输入：6 (n=6)
- 输出：
 

```

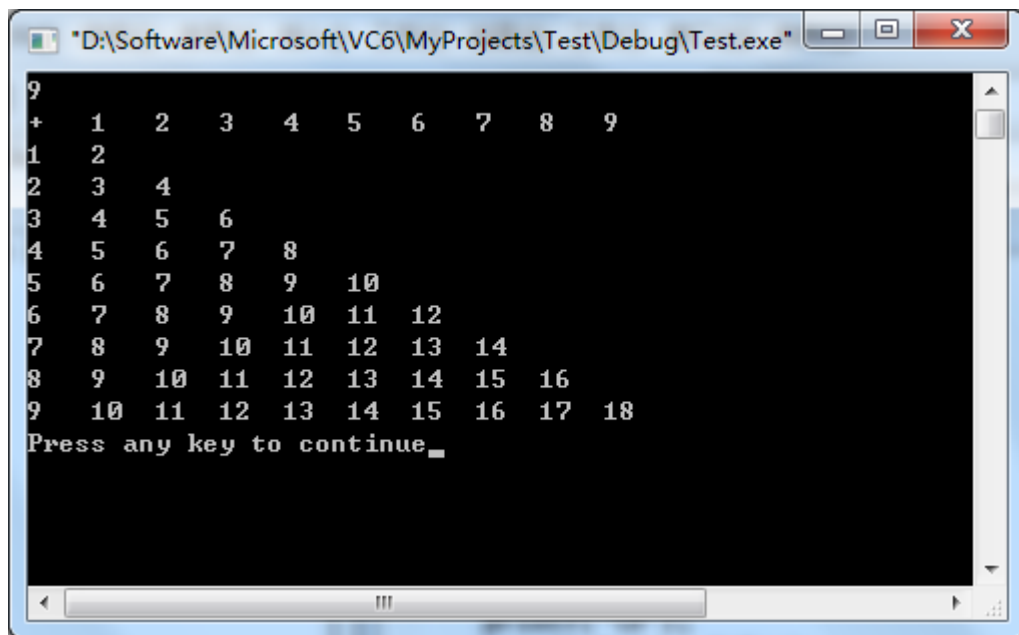
+ 1 2 3 4 5 6
1 2
2 3 4
3 4 5 6

```



```
4 5 6 7 8
5 6 7 8 9 10
6 7 8 9 10 11 12
```

```
#include<stdio.h>
void main()
{
    int n,a[10][10],i,j,m=1;
    scanf("%d",&n);
    printf("+   ");
    while(m<=n)
    {
        printf("%-3d ",m);
        m++;
    }
    printf("\n");
    for (i=0;i<=n-1;i++)
    {
        printf("%-3d ",i+1);
        for (j=0;j<=n-1;j++)
        {
            a[i][j]=i+j+2;
            if(i>=j)
                printf("%-3d ",a[i][j]);
        }
        printf("\n");
    }
}
```



```
D:\Software\Microsoft\VC6\MyProjects\Test\Debug\Test.exe
9
+   1   2   3   4   5   6   7   8   9
1   2
2   3   4
3   4   5   6
4   5   6   7   8
5   6   7   8   9  10
6   7   8   9  10  11  12
7   8   9  10  11  12  13  14
8   9  10  11  12  13  14  15  16
9  10  11  12  13  14  15  16  17  18
Press any key to continue
```

## 7. 字符与字母 （综合题）

用户通过键盘输入一行文档（假设最大长度不超过100个字符），直到按句号“.”键时，输入结束。请编程实现以下的功能：

- (1) 字母的转换：在用户敲键盘的时候，输入的字符，如果它是大写字母，则回显相应的小写字母；如果它是小写字母，则回显相应的大写字母；其他字符，原样回显输出。
- (2) 字符的统计：分别统计出其中的英文字母、空格、数字和其他字符的个数。

例如：

- 输入：（输入时回显）a 19-YEAR-OLD SAYS HE'S READY TO MAKE A DIFFERENCE.  
（用户实际输入为：A 19-year-old says he's ready to make a difference.）
- 输出：（用户按句号“.”后输出，按照规则19-year-old 以及 he's 都被认为是一个词）  
英文字母个数： 37  
空格个数为： 8  
数字个数为： 2  
其他字符个数： 4

（提示，可以使用 `#include <conio.h>` 中的 `getch()` 函数，该函数在键盘输入时，不缓存，不回显）

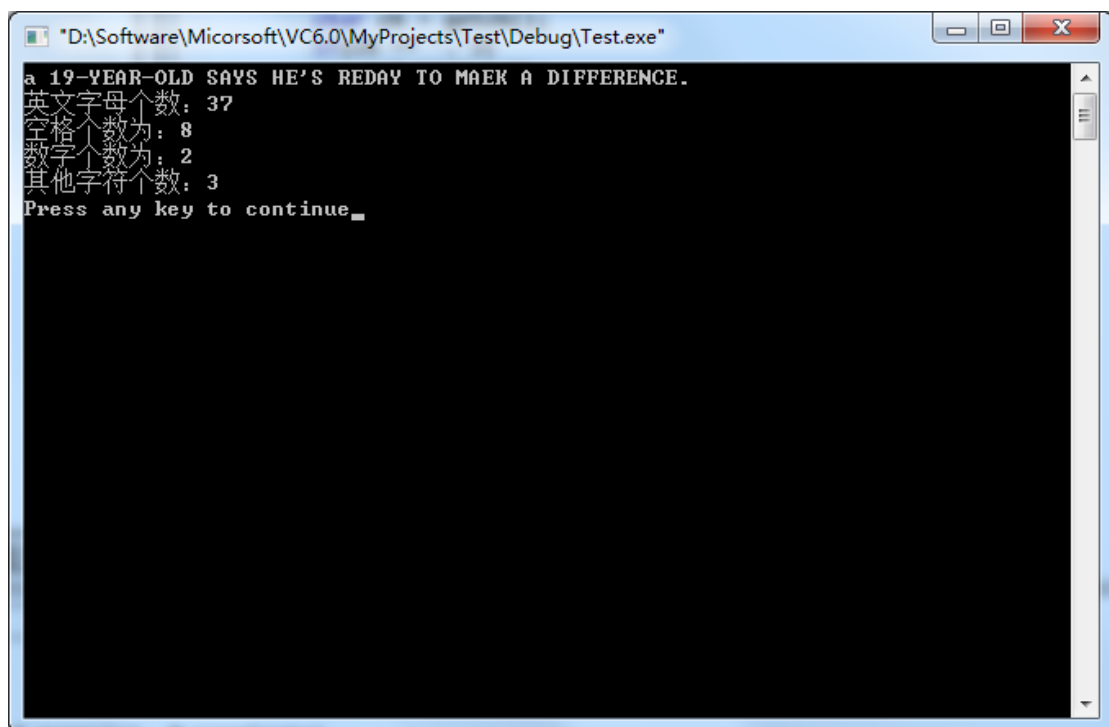
```
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{
    int eng, space, number, other;
    eng = space = number = other = 0;
    for(int i = 0; i < 100; i++){
        char ch = getch();
        if(ch == '.'){
            printf(".\n");
            break;
        }
        else if(ch >='a' && ch <='z'){
            printf("%c", ch-32);
            eng++;
        }
        else if(ch >='A' && ch <='Z'){
            printf("%c", ch+32);
            eng++;
        }
        else if(ch == ' '){
            printf(" ");
            space++;
        }
    }
}
```

```

    }
    else if(ch >= '0' && ch <= '9'){
        printf("%c", ch);
        number++;
    }
    else{
        printf("%c", ch);
        other++;
    }
}

printf("英文字母个数: %d\n空格个数为: %d\n数字个数为: %d\n其他字符个数: %d\n",
eng, space, number, other);
}

```



```

D:\Software\Micorsoft\VC6.0\MyProjects\Test\Debug\Test.exe
a 19-YEAR-OLD SAYS HE'S REDAY TO MAEK A DIFFERENCE.
英文字母个数: 37
空格个数为: 8
数字个数为: 2
其他字符个数: 3
Press any key to continue.

```

## 8. 矩阵的初始化

已知，在 C 语言中，函数 `srand()` 与 `rand()` 配合使用，可以生成一系列的随机数。例如，运行下面的程序，可以生成 10 个 0 到 99 之间的随机数。

```

#include <stdlib.h>
#include <stdio.h>
#include <time.h>

int main(void)
{
    int i;
    printf("Ten random numbers from 0 to 99\n\n");
    srand(time(NULL)); // 以当前的时间为基础，初始化生成随机数的种子
    for(i=0; i<10; i++)
        printf("%d\n", rand() % 100); //rand() 以 srand()为种子，随机生成随机数
    return 0;
}

```

对于一个  $m \times n$  阶的整数型矩阵进行初始化。要求：

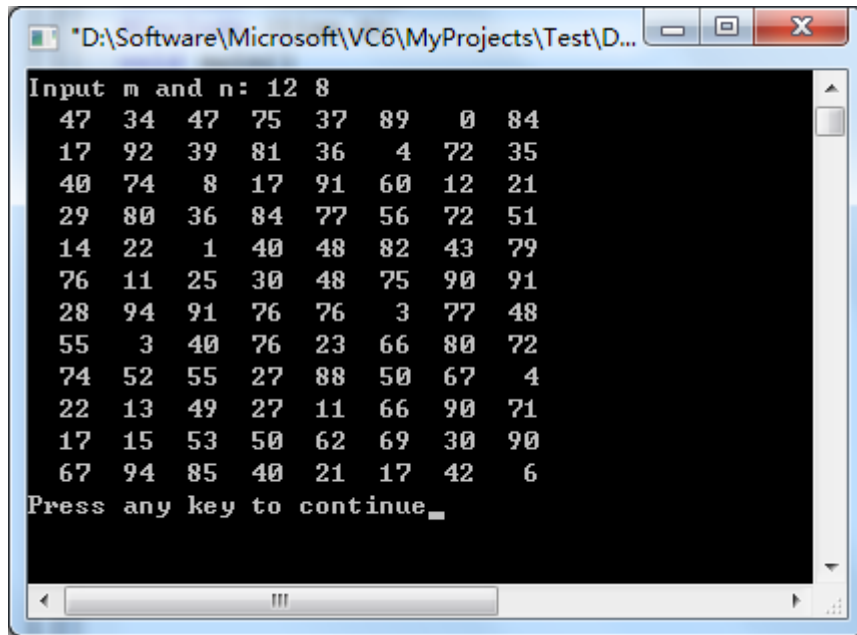
- 用户键盘输入矩阵的阶， $m$  与  $n$  （ $m, n$  在  $0 \sim 15$  之间 如输入  $m=12, n=8$ ）；
- 请使用上面样例提供的随机数生成函数，用  $0 \sim m \times n$  之间的随机数分别初始化该矩阵元素；

在屏幕上输出该矩阵。

```

#include <stdlib.h>
#include <stdio.h>
#include <time.h>
void main()
{
    int m, n, A[15][15];
    printf("Input m and n: ");
    scanf("%d %d", &m, &n);
    srand(time(NULL));
    for(int i = 0; i < m; i++){
        for(int j = 0; j < n; j++){
            A[i][j] = rand()%(m*n);
            printf("%4d", A[i][j]);
        }
        printf("\n");
    }
}

```



## 9. 大数求和（选作）

如你所知，在 C 语言中，基本数据类型中数的表示范围有限（如整数范围不很大）。如何求两个很大的整数（假设整数为 50 位）之和，需要采取其他方法。请编程实现两个大数求和的功能。

例如：

- 输入： 数1 : 112233445566778899  
数2 : 998877665544332211
- 输出： 和为： 111111111111111110

```
#include<stdio.h>
#include<string.h>
#include<conio.h>
void main()
{
    char a[51], b[51], c[52]={'0'}, temp[52];
    int la, lb, lc; //存储三个大数的长度
    int i;
    printf("输入数一: ");
    gets(a);
    printf("输入数二: ");
    gets(b);
    la = strlen(a);
    lb = strlen(b);
    //先对两个输入的大数进行预处理，在前面添加0，使得长度相同
    if(la < lb){
        lc = lb;
        a[lb] = '\0';
```

