

## 实验 7 数组——一维数组及应用

### 一、实验目的

- (1) 掌握一维数组的定义及初始化方法；
- (2) 掌握一维数组中数据的输入和输出方法；
- (3) 掌握与一维数组有关的算法，如查找、插入、排序等；
- (4) 了解用数组处理大量数据时的优越性。

### 二、实验内容和步骤

#### I.基础部分：理解一维数组的存储

- (1) 给定程序 c7-1-1.c，请分析程序的运行结果，回答问题。

```
/*c7-1-1.c*/
#include <stdio.h>
int main()
{
    int a[5]={ 10,20,30,40,50};
    int i;
    printf("下列循环输出每个数组元素的值:\n");
    for(i=0;i<5;i++)
        printf("%5d",a[i]);
    printf("\n 下列循环以十六进制输出每个数组元素的地址:\n");
    for(i=0;i<5;i++)
        printf("%10x",&a[i]);
    printf("\n 下列语句输出数组 a 占的存储空间大小:\n");
    printf("%5d",sizeof(a));
    printf("\n");
    return 0;
}
```

- ① 第二个循环输出一维数组中每个数组元素的地址，观察每 2 个相邻数组元素的地址之差，并由此计算出每一个数组元素所占的存储空间大小；
- ② 通过 `sizeof( )` 可以知道一个简单变量、数组或数据类型所占的空间的大小。例如输出 `sizeof(int)`，得到的数值是 4。语句“`printf("%5d", sizeof(a));`”的输出结果是 20，请分析并说明为什么是 20？

- (2) 某同学希望通过程序 c7-1-2.c 输出数组 a 中所有元素的值。请分析程序的运行结果，回答问题。

```
/*c7-1-2.c*/
#include <stdio.h>
```

```

int main( )
{
    int a[5]={ 10,20,30,40,50};
    printf("输出数组 a 中所有元素的值:\n");
    printf("%5d",a);
    printf("\n");
    return 0;
}

```

- ① 这个程序的输出结果是什么？为什么会输出这个结果？
- ② 如果将语句“printf("%5d",a);”改写成语句“printf("%5d",a+4);”，会输出什么结果？
- ③ 如何修改程序使得程序能够正确输出一维数组 a 中所有元素的值？

## II.提高部分

### 1、程序填空

说明：程序有多个空(1)、(2)……需要补充完整。请将程序中的\_\_ (1) \_\_、\_\_ (2) \_\_……删除后，在相应的位置填入正确答案并调试直到得到正确结果为止。

**注意：不要随意改动程序，不得增行或删行，也不得更改程序的结构！**

(1)给定程序 c7-2-1.c 的功能是：在第一个循环中给 a 数组的前 10 个数组元素依次赋 1、2、3、4、5、6、7、8、9、10；在第二个循环中使 a 数组前 10 个元素中的值对称折叠，变成 1、2、3、4、5、5、4、3、2、1；在最后一个循环中输出对称折叠以后的 a 数组的前 10 个元素。程序运行结果如图 7-1 所示，请把程序补充完整。

数组a:	1	2	3	4	5	6	7	8	9	10
对折后数组a:	1	2	3	4	5	5	4	3	2	1

图 7-1 程序 c7-2-1.c 的运行结果

```

/* c7-2-1.c */
#include <stdio.h>
int main( )
{
    int i, a[10];
    for( i=1; i<=10; i++)
        /*****found*****/
        __ (1) __;
    printf("数组 a:\n");
    for(i=0;i<10;i++)
        printf("%5d",a[i]);
    printf("\n");

    for(i=0;i<5;i++) //对折数组 a 中元素
        /*****found*****/
        __ (2) __;
}

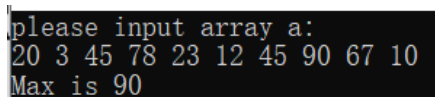
```

```

printf("对折后数组 a: \n");
for(i=0;i<10;i++)
    printf("%5d",a[i]);
printf("\n");
return 0;
}

```

(2) 给定程序 c7-2-2.c 的功能是：把数组 a（大小为 M）中前 M-1 个元素中的最大值放入 a 的最后一个元素中。程序运行结果如图 7-2 所示，请把程序补充完整。



```

please input array a:
20 3 45 78 23 12 45 90 67 10
Max is 90

```

图 7-2 程序 c7-2-2.c 的运行结果

```

/* c7-2-2.c */
#include <stdio.h>
#define M 11
int main()
{
    int a[M],i;
    printf("please input array a:\n");
    for(i=0;i<M-1;i++)
        scanf("%d",&a[i]);
    a[M-1]=a[0];
    /******found******/
    for(i=1; ___ (1) ___;i++)
    /******found******/
        if(___ (2) ___)
            a[M-1]=a[i];
    printf("Max is %d\n",a[M-1]);
    return 0;
}

```

(3) 给定程序 c7-2-3.c 的功能是：实现有序数组的二分查找。程序运行结果示例如图 7-3 所示，请把程序补充完整。



```

请输入待查找的元素值:35
元素35没找到

请输入待查找的元素值:30
元素30的位置为2

```

图 7-3 程序 c7-2-3.c 的运行结果示例

```

/* c7-2-3.c */
#include <stdio.h>
#define N 10
int main()
{
    int a[N]={ 10,20,30,40,50,60,70,80,90,100};
    int low,high,mid,x;

```

```

printf("请输入待查找的元素值:");
scanf("%d",&x);
low=0;
/*****found*****/
high=____(1)____;
/*****found*****/
while(____ (2) ____ )
{
    mid=(low+high)/2;
    if(x==a[mid] )
    {
        printf("元素%d 的位置为%d\n",x,mid);
        return 1;
    }
    else
/*****found*****/
        ____ (3) ____;
        else low=mid+1;
    }
printf("元素%d 没找到\n",x);
return 0;
}

```

## 2、程序改错

说明：程序中有错误，错误都在提示行： /\*\*\*\*\*found\*\*\*\*\*/的下面一行。请改正程序中的错误，编译运行使它能得出正确的结果。

**注意：程序中的其它地方不要随意改动，不得增行或删除行，也不得更改程序的结构！**

(1) 给定程序 c7-2-4.c 的功能是：求一维数组 a 中所有元素的平均值，结果保留两位小数。程序的运行结果如图 7-4 所示，请改正程序中的错误，并运行出正确的结果。

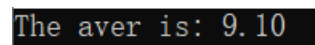


图 7-4 程序 c7-2-4.c 的运行结果示例

```

/*  c7-2-4.c  */
#include <stdio.h>
int main()
{
/*****found*****/
    int a (10) = (10,4,2,7,3,12,5,34,5,9) , i;
    float  aver, s;
/*****found*****/
    s = 0;
    for ( i=1; i<10; i++)

```

```

        s = s+a[i];
aver = s / i;
printf("The aver is: %.2f\n", aver);
return 0;
}

```

(2) 给定程序 c7-2-5.c 的功能是：输入一个数 x，查找 x 在数组 a 中最后一次出现的位置。程序的运行结果如图 7-5 所示，请改正程序中的错误，并运行出正确的结果。



```

please input x:5
last 5 is No. 3

please input x:10
10 not found !

```

图 7-5 程序 c7-2-4.c 的运行结果示例

```

/* c7-2-5.c */
#include <stdio.h>
int main()
{
    int a[10]={ 1, 5, 2, 5, 6, 8, 7, 4, 3, 0 };
    int i,k, x,f=0;
    printf("please input x:");
    scanf("%d",&x);
    for(i=0;i<10;i++)
        if(a[i]== x )
        {
            f=1;
            /*****found*****/
            k=i;
        }
    /*****found*****/
    if( f=1)
        printf("last %d is No. %d\n", x,k);
    else
        printf(" %d not found !\n", x);
    return 0;
}

```

### 3、程序设计

(1) 编写程序 c7-2-6.c，实现的功能是：求一维数组中下标为偶数的元素之和。

#### 编程提示：

- 定义一个数组 a 并输入数据或初始化。
- 定义一个整型变量 sum，存放下标为偶数的元素的和，并初始化为 0。
- 从数组的第 0 个元素开始，每次循环控制变量递增 2，直到处理至数组的最后一个元素，将满足条件的数组元素值累加到 sum 变量中。
- 输出 sum 变量的值，即求出下标为偶数的元素之和。

(2) 编写程序 c7-2-7.c，实现的功能是：在一个已经从小到大排好序的整数数组中，例如：a[20]={6, 9, 12, 15, 19, 23, 45, 67, 89, 98}，从键盘输入一个整数 x，将 x 插入到数组中，插

入后的数组仍然保持升序。

(3) 编写程序 **c7-2-8.c**, 实现的功能是: 已知一个数组, 数组中的元素各不相同, 从键盘输入一个数  $x$ , 如果数组中存在  $x$ , 则从数组中删除它, 如果数组中不存在  $x$ , 则给出相应信息。

(4) 编写程序 **c7-2-9.c**, 实现的功能是: 已知一个数组, 数组中元素无序, 而且数组中的元素可能相同。从键盘输入一个数  $x$ , 从数组中删除所有和  $x$  相同的元素。

例如: 假设有数组  $a[10]=\{6, 9, 12, 9, 8, 20, 9, 6, 15, 5\}$ , 输入  $x=9$ , 删除后的数组元素为 6, 12, 8, 20, 6, 15, 5。

(5) 编写程序 **c7-2-10.c**, 实现的功能是: 将 100 以内的素数存放到一个数组中。

**编程提示:**

- 首先理解和掌握判断素数的算法。
- 定义一个数组存放 100 以内的素数, 想一想该数组的大小应该为多少?
- 定义两个整型变量分别作内外层循环的控制变量。
- 定义一个整型变量作为数组元素下标的计数器, 想一想该变量应赋什么样的初值?
- 利用双重循环, 外层循环对 1~100 之间的所有整数依次进行判断; 内层循环则判断每个整数是否为素数。如果是素数, 存放 to 数组中, 并使数组下标变量加 1; 否则继续判断下一个整数。
- 利用用循环语句输出数组中的所有素数, 注意循环变量的初值和终值如何确定。

(6) 编写程序 **c7-2-11.c**, 实现的功能是: 模拟骰子的 6000 次投掷, 统计骰子的 6 个面各自出现的概率。

**编程提示:**


- 假设筛子的 6 个面分别用数字 1-6 表示, 每次投掷的结果就是用随机数产生 1 个 1-6 之间的随机数。
- 6 个面各自出现的次数保存到一个一维数组中。

### III. 拓展部分: 利用一维数组解决大数存储问题

(1) 编写程序 **c7-3.c**, 挑战类型表示的极限——大数存储和运算 (V1.0 版)

C 语言中无论是 `int` 还是 `long int` 类型, 数据的存储范围都是有限的, 常用的整型数据在内存中占用四个字节, 能表示的数据范围是  $-2147483648 \sim 2147483647$ , 即使在 C++ 中, 最大的 `long long` 数据类型也只能存储  $-9223372036854775808 \sim 9223372036854775807$  范围内的数据, 当整数的值超过这个存储范围, 将不能被正确表示。

一种解决方法是用一个 `int` 类型数组存放和表示大数, 一个数组元素存放大数中的一位。例如: 定义一个 `int A[200]`, 可以存放一个最多为 200 位的无符号整数。假设有一个大整数 123456789123456789, 保存到 A 数组中, 数组中的数据表示如下:

																			下标
19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	.....	0	0	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7
8	9																		

编程实现 2 个大整数 A 和 B 的相加, 注意相加过程中可能会出现进位, 相加结果保存到另一个大整数 C 中。

## 实验 8 数组—二维数组及应用

### 一、实验目的

- (1) 掌握二维数组的定义及初始化方法；
- (2) 掌握二维数组中数据的输入和输出方法；
- (3) 掌握与二维数组有关的算法。

### 二、实验内容和步骤

#### I.基础部分：理解二维数组的存储

- (1) 给定程序 c8-1-1.c, 请分析程序的运行结果, 回答问题。

```
/* c8-1-1.c */
#include <stdio.h>
#define M 3
#define N 5
int main()
{
    char a[M][N]={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O'};
    int i,j;
    printf("下列循环输出每个数组元素的值:\n");
    for(i=0;i<M;i++)
    {
        for(j=0;j<N;j++)
            printf("%5c",a[i][j]);
        printf("\n");
    }
    printf("\n 下列循环以十六进制输出每个数组元素的地址:\n");
    for(i=0;i<M;i++)
    {
        for(j=0;j<N;j++)
            printf("%10x",&a[i][j]);
        printf("\n");
    }
    printf("\n 下列语句输出数组 a 占的存储空间大小:\n");
    printf("%5d",sizeof(a));
    printf("\n");
    return 0;
}
```

- ① 第二个循环输出每个数组元素的地址, 观察运行结果分析每 2 个相邻数组元素的地址之差, 并由此计算出每一个数组元素所占的存储空间大小;
- ② 从每个元素的存储地址看, 数组 a 中的元素在内存中是按照先行后列的顺序存储, 还是

先列后行的顺序存储的？

- ③ 最后一个输出语句“`printf("%5d", sizeof(a));`”的输出结果是 15，请说明为什么是 15？

(2) 某同学希望通过下列程序 `c8-1-2.c` 输出二维数组 `a` 中所有元素的值。请分析程序的运行结果，回答问题。

```
/* c8-1-2.c */
#include <stdio.h>
int main( )
{
    int a[3][3]={ 10,20,30,40,50,60};
    printf("输出数组 a 中所有元素的值:\n");
    printf("%5d",a);
    printf("\n");
    return 0;
}
```

- ① 这个程序的输出结果是什么？为什么会输出这个结果？
- ② 如果将语句“`printf("%5d",a);`”改写成语句“`printf("%5d",a[1]);`”，会输出什么结果？为什么会输出这个结果？
- ③ 如果将语句“`printf("%5d",a);`”改写成语句“`printf("%5d",a[1][0]);`”，会输出什么结果？为什么会输出这个结果？
- ④ 如何修改程序使得程序能够正确输出二维数组 `a` 中所有元素的值？

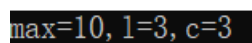
## II.提高部分

### 1、程序填空

说明：程序有多个空(1)、(2)……需要补充完整。请将程序中的\_\_ (1) \_\_、\_\_ (2) \_\_……删除后，在相应的位置填入正确答案并调试直到得到正确结果为止。

**注意：不要随意改动程序，不得增行或删行，也不得更改程序的结构！**

(1) 给定程序 `c8-2-1.c` 的功能是：求二维数组 `a` 中的最大元素及其下标。程序运行结果示例如图 8-1 所示，请把程序补充完整。



max=10, l=3, c=3

图 8-1 程序 `c8-2-1.c` 的运行结果

```
/* c8-2-1.c */
#include "stdio.h"
int main( )
{
    int a[4][4]={ { 1,2,3,4},{ 3,4,5,6},{ 5,6,7,8},{ 7,8,9,10}};
    int i,j,max,l,c;
    /*******found*****/
    max=____ (1) ____;
    for(i=0; i<4; i++)
        for(j=0; j<4; j++)
```



```

/*****found*****/
    if( ____ (2) ____ )
    {
        max=a[i][j];
        l=i;
        c=j;
    }
    printf("max=%d,l=%d,c=%d\n",max,l,c);
    return 0;
}

```

(2)给定程序 c8-2-2.c的功能是：自动形成并输出如图 8-2 所示的矩阵,请把程序补充完整。

```

array a:
1 2 3 4 5
1 1 6 7 8
1 1 1 9 10
1 1 1 1 11
1 1 1 1 1

```

图 8-2 程序 c8-2-2.c 的运行结果

```

/* c8-2-2.c */
#include <stdio.h>
int main()
{
    int i,j,k,a[5][5];
    k=2;
    for(i=0; i<5; i++) /* 行循环 */
        for(j=0; j<5; j++) /* 列循环 */
            /*****found*****/
            if( ____ (1) ____ )
                a[i][j]=1; /* 产生矩阵的下三角元素 */
            else
                a[i][j]=k++; /* 产生矩阵的上三角元素 */
    printf("array a:\n");
    for(i=0; i<5; i++)
    {
        for(j=0; j<5; j++)
            printf("%4d",a[i][j]);
        /*****found*****/
        ____ (2) ____; /* 每输出一行后换行 */
    }
    return 0;
}

```

## 2、程序改错

说明：程序中有错误，错误都在提示行： /\*\*\*\*\*found\*\*\*\*\*/的下面一行。请改正程序中的错误，编译运行使它得出正确的结果。

**注意：程序中的其它地方不要随意改动，不得增行或删行，也不得更改程序的结构！**

(1) 给定程序 c8-2-3.c 的功能是：求二维数组 a 中每行的最小值元素，结果保存在一维数组 min 中。程序的运行结果如图 8-3 所示，请改正程序中的错误，并运行出正确的结果。

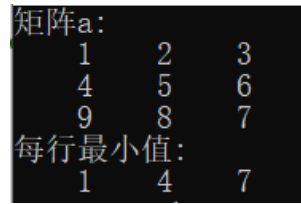


Figure 8-3 shows the output of the program. It displays a 3x3 matrix 'a' with values 1, 2, 3 in the first row; 4, 5, 6 in the second row; and 9, 8, 7 in the third row. Below the matrix, it shows the row minimums: 1, 4, 7.

矩阵a:			
1	2	3	
4	5	6	
9	8	7	

每行最小值:			
1	4	7	

图 8-3 程序 c8-2-3.c 的运行结果

```
/* c8-2-3.c */
#include <stdio.h>
int main()
{
    int a[3][3]={1,2,3,4,5,6,9,8,7}, min[3],i,j;
    for(i=0;i<=2;i++)
    {
        /******found******/
        min[i]=a[0][0];
        for ( j=1; j<=2; j++)
        /******found******/
            if(min[i]<a[i][j])
                min[i]=a[i][j];
    }
    printf("矩阵 a:\n");
    for(i=0;i<=2;i++)
    {
        for ( j=0; j<=2; j++)
            printf("%5d", a[i][j] );
        printf("\n");
    }
    printf("每行最小值:\n");
    /******found******/
    for(i=1;i<=3;i++)
        printf("%5d",min[i]);
    printf("\n");
    return 0;
}
```

(2) 给定程序 c8-2-4.c 的功能是：实现矩阵的转置。程序的运行结果如图 8-4 所示，请改正程序中的错误，并运行出正确的结果。

```

矩阵a:
  1   2   3
  4   5   6
  7   8   9
转置后矩阵:
  1   4   7
  2   5   8
  3   6   9

```

图 8-4 程序 c8-2-4.c 的运行结果示例

```

/*  c8-2-4.c  */
#include <stdio.h>
int main()
{
    /******found******/
    int a[][]={1,2,3,4,5,6,7,8,9}, i,j,t;
    printf("矩阵 a:\n");
    for(i=0;i<=2;i++)
    {
        for ( j=0; j<=2; j++)
            printf("%5d", a[i][j] );
        printf("\n");
    }
    for(i=0;i<3;i++)
    /******found******/
        for(j=0;j<3;j++)
        {
            t=a[i][j];
            /******found******/
            a[j][i] = a[i][j];
            a[j][i]=t;
        }
    printf("转置后矩阵:\n");
    for(i=0;i<=2;i++)
    {
        for ( j=0; j<=2; j++)
            printf("%5d", a[i][j] );
        printf("\n");
    }
    return 0;
}

```

### 3、程序设计

(1) 编写程序 **c8-2-5.c**，实现的功能是：打印输出以下的杨辉三角（要求打印出 10 行）

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

#### 编程提示：

- 杨辉三角的特点是：第 1 列和对角线上的元素为 1，其它各元素的值都是上一行上一列元素和上一行前一列元素之和。
- 定义一个  $10 \times 10$  的二维数组 **a**。
- 先用一个单层循环为第一列和对角线上的元素赋值。
- 再用一个双重循环为其它元素赋值。
- 最后再用一个双重循环输出杨辉三角。注意各层循环变量的初值和终值怎样确定。

(2) 编写程序 **c8-2-6.c**，输入一个 N 阶方阵，判断该方阵是否对称。

(3) 编写程序 **c8-2-7.c**，求两个矩阵的积。

### III.拓展部分：利用二维数组解决实际问题

编写大奖赛现场计分程序 **c8-3.c**：已知某大奖赛有  $n$  ( $n > 1$ ) 个参赛选手， $m$  ( $m > 2$ ) 个评委为参赛选手评分（最高 10 分，最低 0 分）。统分规则为：在每个选手的  $m$  个得分中，去掉一个最高分和一个最低分，计算平均分。根据  $n$  个选手的最后得分，从高到低输出选手的得分名次表（可以以追加数据的方式输出至文件 `score.txt` 中），以确定获奖名单。

已知选手的编号及  $m$  个得分（假设  $m=5$ ）已经保存在文件 `score.txt` 中，格式如下：

```
101  9.0  8.5  9.5  8.0  8.5
102  8.0  8.5  9.0  9.0  8.0
103  9.5  9.5  10.0  9.5  9.5
104  7.0  6.5  6.5  7.0  7.5
105  9.0  9.0  9.5  9.5  9.5
```

.....

## 实验 9 数组—字符数组、字符串及应用

### 一、实验目的

- (1) 掌握字符数组和字符串的概念；
- (2) 掌握字符串函数的使用；
- (3) 了解字符串数据的处理及应用。

### 二、实验内容和步骤

#### I. 基础部分: 理解字符数组和字符串

- (1) 给定程序 c9-1-1.c, 请分析程序的运行结果, 回答问题。

```
/* c9-1-1.c */
#include <stdio.h>
int main( )
{
    char ch[ ]={'c','o','m','p','u','t','e','r'};
    int i;
    for(i=0;i<8;i++)
        printf("%c",ch[i]);
    printf("\n");
    printf("数组 ch 的大小为: %d\n",sizeof(ch)/sizeof(char));
    return 0;
}
```

- ① 最后一个 printf 语句可以输出数组 ch 的大小, 输出的大小是否和数组元素的个数一致?
- ② 若将程序中的第 1 行 “char ch[ ]={'c','o','m','p','u','t','e','r'};” 改为: “char ch[]={“computer”};”, 如程序 c9-1-2.c 中所示, 最后一个 printf 语句输出的数组 ch 的大小是否和 c9-1-1.c 中数组的大小一致? 为什么?

```
/* c9-1-2.c */
#include "stdio.h"
int main( )
{
    char ch[]={“computer”};
    int i;
    for(i=0;i<8;i++)
        printf("%c",ch[i]);
    printf("\n");
    printf("数组 ch 的大小为: %d\n", sizeof(ch)/sizeof(char));
}
```

```

    return 0;
}

```

- ③ 修改程序 c9-1-2.c, 在程序最后增加语句: `printf("字符串 ch 的长度为%d\n",strlen(ch));` 如程序 c9-1-3.c 中所示。程序中最后两个 `printf` 语句分别输出的数组 `ch` 的大小和字符串 `ch` 的长度, 这两个输出结果为啥不一致?

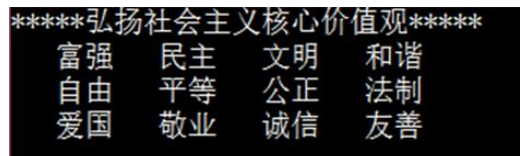
```

/* c9-1-3.c */
#include<string.h>
#include "stdio.h"
int main()
{
    char ch[]={ "computer" };
    int i;
    for(i=0;ch[i]!='\0';i++)
        printf("%c",ch[i]);
    printf("\n");
    printf("数组 ch 的大小为: %d\n", sizeof(ch)/sizeof(char));
    printf("字符串 ch 的长度为%d\n",strlen(ch));
    return 0;
}

```

- ④ 将程序 c9-1-3.c 中的 “`printf("%c",ch[i]);`” 改为: “`printf("%s",ch) ;`”, 分析程序的输出结果。

(2) 给定程序 c9-1-4.c, 程序的运行结果如图 10-1 所示, 请分析程序的运行结果, 回答问题。



```

*****弘扬社会主义核心价值观*****
富强  民主  文明  和谐
自由  平等  公正  法制
爱国  敬业  诚信  友善

```

图 10-1 程序 c9-1-4.c 的运行结果

```

/* c9-1-4.c */
#include <stdio.h>
int main()
{
    char concept[12][20]={ "富强","民主","文明","和谐","自由","平等","公正","法制","爱
    国","敬业","诚信","友善" };
    int i;
    printf("*****弘扬社会主义核心价值观*****\n");
    for(i=0;i<12;i++)
    {
        printf("    %s",concept[i]);
        if((i+1)%4==0) printf("\n");
    }
}

```

```

    return 0;
}

```

①请在下图的二维数组存储空间中填入数组 `concept` 的对应内容。

下 标	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0																				
1																				
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
10																				
11																				

② 程序中，数组 `concept` 是二维数组，但是 `printf("%s", concept[i])` 语句中，使用的是 `concept[i]`。`concept[i]`代表的含义是什么？

## II.提高部分

### 1、程序填空

说明：程序有多个空 (1)、(2) ……需要补充完整。请将程序中的\_\_ (1) \_\_、\_\_ (2) \_\_ ……删除后，在相应的位置填入正确答案并调试直到得到正确结果为止。

**注意：不要随意改动程序，不得增行或删行，也不得更改程序的结构！**

(1) 给定程序 `c9-2-1.c` 的功能是：从键盘上输入一个数字字符组成的字符串（字符串长度小于 9），将该字符串转换成一个十进制数。程序运行结果示例如图 10-1 所示，请把程序补充完整。



图 10-2 程序 `c9-2-1.c` 的运行结果示例

```

/*   c9-2-1.c   */
#include <stdio.h>
int main()
{
    char  s[10]; int i;
    int  n=0;
    printf("请输入数字字符串:");
    gets(s);

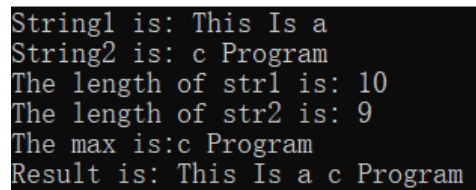
```

```

/*****found*****/
for(i=0; ____ (1) ____; i++)
{
/*****found*****/
    if (____ (2) ____ )
    {
        printf("输入非法字符\n");
        return 1;
    }
    else
/*****found*****/
        n = n*10 + ____ (3) ____;
    }
    printf("n=%ld\n",n);
    return 0;
}

```

(2) 给定程序 c9-2-2.c 的功能是：用 strlen 函数分别测量字符串 str1 和 str2 的长度，用 strcmp 函数比较两个字符串的大小，最后用 strcat 函数实现将字符串 str2 连接到字符串 str1 的后面并输出。程序运行结果示例如图 10-2 所示，请将程序补充完整。



```

String1 is: This Is a
String2 is: c Program
The length of str1 is: 10
The length of str2 is: 9
The max is:c Program
Result is: This Is a c Program

```

图 10-2 程序 c9-2-2.c 的运行结果示例

```

/* c9-2-2.c */
#include <stdio.h>
#include<string.h>
int main( )
{
    char str1[80]="This Is a ",str2[80]="c Program";
    printf("String1 is: %s\n",str1);
    printf("String2 is: %s\n",str2);
/*****found*****/
    printf("The length of str1 is: %d\n",____ (1) ____); /*使用 strlen 函数实现*/
    printf("The length of str2 is: %d\n", strlen(str2));
/*****found*****/
    if( ____ (2) ____ ) /*使用 strcmp 函数实现比较字符串大小*/
        printf("The max is:%s\n",str1);
    else
        printf("The max is:%s\n",str2);
/*****found*****/
}

```



```

_____(3)_____*/*使用 strcat 函数实现字符串连接*/
printf("Result is: %s\n",str1);
return 0;
}

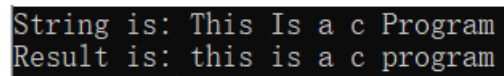
```

## 2、程序改错

说明：程序中有错误，错误都在提示行： `/******found******/` 的下面一行。请改正程序中的错误，编译运行使它能得出正确的结果。

**注意：程序中的其它地方不要随意改动，不得增行或删行，也不得更改程序的结构！**

(1) 给定程序 c9-2-3.c 的功能是：实现将一个字符串中的所有大写字母转换为小写字母并输出。程序的运行结果如图 10-3 所示，请改正程序中的错误，并运行出正确的结果。



```

String is: This Is a c Program
Result is: this is a c program

```

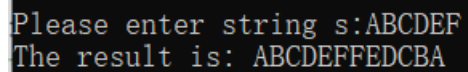
图 10-3 程序 c9-2-3.c 的运行结果示例

```

/* c9-2-3.c */
#include <stdio.h>
int main()
{
    char str[80]="This Is a c Program";
    int i;
    printf("String is: %s\n", str);
    /******found******/
    for (i=0; str[i]!='\0'; i++)
        if (str[i]>='A' && str[i]<='Z')
            /******found******/
            str[i]=str[i]-32;
    /******found******/
    printf("Result is: %c\n",str);
    return 0;
}

```

(2) 给定程序 c9-2-4.c 的功能是：先将字符串 s 中的字符按正序存放到 t 字符串中，然后把 s 中的字符按逆序连接到 t 字符串的后面。程序的运行结果如图 10-4 所示，请改正程序中的错误，并运行出正确的结果。



```

Please enter string s:ABCDEF
The result is: ABCDEFFEDCBA

```

图 10-4 程序 c9-2-4.c 的运行结果示例

```

/* c9-2-4.c */
#include <stdio.h>
#include <string.h>
int main()
{

```

```

char s[80],t[80];
int i, len;
printf("\nPlease enter string s:");
/*****found*****/
scanf("%s", &s);
len = strlen(s);
/*****found*****/
for(i=0; i<=len; i++)
    t[i] = s[i];
for(i=0; i<len; i++)
/*****found*****/
    t[len+i] = s[len-i];
/*****found*****/
t[2*len-1] =0;
printf("The result is: %s\n", t);
return 0;
}

```

### 3、程序设计

(1) 编写程序 c9-2-5.c, 实现的功能是: 在一个字符串中查找指定的字符, 并输出指定的字符在字符串中出现的次数及位置, 如果该字符串中不包含指定的字符, 请输出提示信息。

#### 编程提示:

- 定义两个一维数组, a 是字符数组用来存放字符串, b 是整数数组用来存放指定的字符在字符串中出现的位置(即对应的下标)。
- 定义 i, j, m 三个循环控制变量和一个标志变量 flag, 并初始化 flag 的值为 0。
- scanf 或者 gets 函数输入一个字符串。
- 在循环中对字符数组的每个元素和指定字符 ch 进行匹配判断, 如果相同, 就把其下标依次存放在数组 b 中, 并置 flag 的值为 1。
- 循环退出后判断标志变量 flag 的值, 如果仍为 0, 说明字符串中没出现指定的字符, 否则, 就输出该字符在字符串中出现的次数和位置。

(2) 编写程序 c9-2-6.c, 实现的功能是: 对从键盘输入的一行文字(不超过 80 个字符), 分别统计出其中的英文大写字母、小写字母、数字、空格、及其它字符的个数。

#### 编程提示:

- 用一维字符数组存放字符串。
- 定义 5 个整型变量分别统计大写字母、小写字母、数字、空格和其它字符的个数(即作为 5 个计数器使用), 并为这 5 个变量赋初值。
- 用 scanf 函数或 gets 函数为字符数组赋一个字符串。
- 在循环中对字符数组的每个元素进行判断, 相应的计数器加 1。注意循环控制的条件和进行判断的条件怎样设置。
- 循环结束后输出各计数器的值。

(3) 编写程序 c9-2-7.c, 实现的功能是: 将输入的一个字符串逆序存放后输出。

例如, 当字符串为 "This Is a c Program"

输出为: "margorP c a sI sihT"

#### 编程提示:

- 定义一个字符数组，用于输入字符串；
- 实现将字符数组中的元素逆序存放时，判断字符串处理完毕可以使用 ‘\0’ 或 strlen 函数
- 程序的参考结构如下：

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

int main()
{
    定义字符串及相应的变量；
    调用相关函数从键盘输入字符串至字符数组中；
    访问字符数组中的每个元素，并换到相应的位置；
    输出字符串；
}
```

(4) 编写程序 c9-2-8.c，实现的功能是：将两个字符串连接起来并输出结果，注意不使用 strcat 函数。请完成程序。

#### 编程提示：

- 定义两个一维字符型数组 str1、str2 和两个循环变量。
- 为两个字符数组输入两个字符串（可使用 scanf 函数或 gets 函数整体赋值，要注意 scanf 和 gets 函数的区别，在对字符串赋值时，scanf 函数不能出现空格）。
- 确定字符数组 str1 结束的位置。
- 再将字符数组 str2 中的内容连接到字符数组 str1 的后面。
- 为字符数组 str1 赋字符串结束的标志 ‘\0’。
- 输出连接后的字符数组 str1。
- 程序的参考结构如下

```
#include <stdio.h>

int main( )
{
    定义字符数组存放字符串；
    定义必要的循环计数变量；
    分别输入两个字符串；
    在第一个字符串中查找要串接的位置；
    将第二个字符串串接至第一个字符串相应的位置；
    给出新的字符串的结束符；
    printf("the catenated string is %s", str1);
    return 0;
}
```

### III.拓展部分：字符数组和字符串的应用

#### (1) 学生成绩管理系统 (V2.0 版)

文件 score.txt 中存放若干同学的学号、姓名及高数、英语、C 语言 3 门课的成绩，格式如下：

```
1001  张晓莉   90  80  70
1002  赵  龙   85  78  80
1003  李大有   60  70  76
```

.....

编写程序 c9-3-1.c，从 score.txt 中逐个读入每个同学的成绩，要求：

- ① 计算平均分
- ② 成绩按平均分排序
- ③ ③屏幕输出结果并写入文件 score\_avg.txt 中。

文件 score\_avg.txt 中的记录应该为：

学号	姓名	高数	英语	C 语言	平均分
1002	赵 龙	85	78	80	81.00
1001	张晓莉	90	80	70	80.00
1003	李大有	60	70	76	68.67

.....

(2) 程序 c9-3-2.c 实现的功能是：在一个文本文档中查找特定字符串。

要求：

在文本文档“使命与担当.txt”中查找指定关键字的一句话，如：输入关键字“新时代的青年”，则将含有该关键字的一句话，复制粘贴到新的文本文档“学习笔记.txt”中。

### 文本文档“使命与担当.txt”

题目：新时代青年的使命与担当

随着中国特色社会主义进入新时代，青年的成长成才也产生了新的历史方位和环境条件。青年人具有知识、观念和创新优势，在推进新时代改革开放事业、实施“四个伟大”的过程中，必须充分发挥青年群体作为建设者的积极作用；青年人具有体能、技能和智能优势，这一优势为青年担当起实现“两个一百年”奋斗目标的任务、成为新时代中国特色社会主义事业的接班人提供了必要条件。历史赋予使命，时代要求担当。习近平总书记在纪念五四运动 100 周年大会上号召：“新时代中国青年要珍惜这个时代、担负时代使命，在担当中历练，在尽责中成长。”，并对青年担当尽责、成长成才提出了六点期望和要求。对照习近平总书记的期望和要求，青年人要切实承担起推进新时代中国特色社会主义事业的使命，努力成长为新时代德智体美劳全面发展的社会主义建设者和接班人。作为新时代的青年，应志存高远、忠于祖国，努力做新时代具有远大理想和坚定信念的爱国者。“志不立，天下无可成之事”。习近平总书记勉励广大青年“要励志，立鸿鹄志”，并指出，“热爱祖国是立身之本、成才之基”，是“立德之源、立功之本”。一个人的理想只有同国家的前途和民族的命运相结合才有价值，一个人的追求只有同社会的需要和人民的利益相一致才有意义。新时代青年只有胸怀忧国忧民之心、爱国爱民之情，才能准确定位自己的人生目标和奋斗目标。“信念决定事业成败。没有理想信念，就会导致精神上‘缺钙’”。新时代青年只有用习近平新时代中国特色社会主义思想武装头脑，不断增强“四个自信”、持续坚定中国特色社会主义信念，才能在推进新时代中国特色社会主义事业的爱国奋斗中不断实现人生理想和价值。作为新时代的青年，应敢于担当、勇于奋斗，努力做新时代具有责任意识和创新精神的建设者。国家命运与个人前途休戚相关，民族振兴与个体发展紧密相连。习近平总书记指出，“新时代中国青年处在中华民族发展的最好时期，既面临着难得的建功立业的人生际遇，也面临着‘天将降大任于斯人’的时代使命”，希望“新时代中国青年要担当时代责任”。在中国迎来从站起来、富起来到强起来的伟大飞跃新时代，广大青年应深刻认识自身所面临的时代际遇和历史责任，将个人梦与中国梦结合起来，以实现中华民族伟大复兴为己任，不辜负党的期望、人民期待和民族重托，不断将中国特色社会主义事业推向前进。但是，中华民族伟大复兴绝不是轻轻松松、敲锣打鼓就能实现的，需要靠一代又一代人的接续奋斗。“奋斗是青春最亮丽的底色”，广大青年要积极响应习近平总书记的号召，“青春是用来奋斗的”，要有“锐意创新的勇气、敢为人先的锐气、蓬勃向上的朝气”，“勇于创业、敢闯敢干，努力在改革开放中闯新路、创新业态，不断开辟事业发展新天地”。作为新时代的青年，应勤奋学习、锤炼身心，努力做新

时代具有过硬本领和高尚品格的接班人。习近平总书记教育广大青年：“梦想从学习开始、事业靠本领成就。”追求梦想、担当使命需要依靠过硬的本领，而练就过硬本领则要依靠勤奋学习。青年时期是学习的黄金期，青年要把学习作为首要任务，不仅要学书本上的知识，更要学实践中的知识。要在面向现代化、面向世界、面向未来的大局中不断提升体能、技能和智能，要在感悟新时代、紧跟新时代、引领新时代的新际遇中持续提高自身的素质和能力，通过学习使自己成为新知识、新观念和新思维的集成体。与此同时，要注重修炼品德。新时代青年要不断用社会主义核心价值观涵养自身的言行品格，自觉按照党和人民要求不断锤炼自己、完善自己。自身的提高是为了成为建设国家的有用之材，而这一价值的最终体现则要通过实践来实现，广大青年要积极投身于新时代中国特色社会主义的伟大实践，努力在新时代改革开放事业的奋斗中成为可堪大用、能担重任的栋梁之材。

### 学习笔记 1.txt（当输入“新时代的青年”）

作为新时代的青年，应志存高远、忠于祖国，努力做新时代具有远大理想和坚定信念的爱国者。

作为新时代的青年，应敢于担当、勇于奋斗，努力做新时代具有责任意识和创新精神的建设者。

作为新时代的青年，应勤奋学习、锤炼身心，努力做新时代具有过硬本领和高尚品格的接班人。

### 学习笔记 2.txt（当输入“新时代青年”）

新时代青年的使命与担当

新时代青年只有胸怀忧国忧民之心、爱国爱民之情，才能准确定位自己的人生目标和奋斗方向。“信念决定事业成败。没有理想信念，就会导致精神上‘缺钙’”。

新时代青年只有用习近平新时代中国特色社会主义思想武装头脑，不断增强“四个自信”、持续坚定中国特色社会主义信念，才能在推进新时代中国特色社会主义事业的爱国奋斗中不断实现人生理想和价值。

新时代青年要不断用社会主义核心价值观涵养自身的言行品格，自觉按照党和人民要求不断锤炼自己、完善自己。