**计算机程序设计基础（C++)**

**实验报告**

专业班级： 软工2306班

学 号： 8210230324

姓 名： 徐梓程

**实验报告成绩：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验** | **实验一** | **实验二** | **实验三** | **实验四** | **实验五** | **总评** |
| **成绩** |  |  |  |  |  |  |

**批阅教师：**

**实验三 函数**

**一、实验目的**

本实验主要培养、训练学生对函数的理解，要求：

1. 掌握函数的定义、声明的方法；

2. 掌握函数的编写要求；

3. 掌握函数的调用方法；

4. 掌握函数参数的传递方法；

5. 掌握变量的作用域；

6. 掌握多文件编程方法。

**二、实验内容与要求**

1、输入自然数m和n，

（1）求他们的最大公约数（或称最大公因数）。

要求输入、输出在主函数中进行，求公约数由函数实现。

1. 在函数中求最大公约数与最小公倍数。（提示：使用引用参数）

2. 编写程序满足：声明一个函数，判断一个整数是否为素数，使用如下函数头：

bool is\_prime(int num) ,如果num是素数函数返回true，否则返回false；

利用函数is\_prime找出前200个素数，并按每行10个输出：

     2     3      5      7    11    13    17    19    23    29

3、编程实现摄氏温度到华氏温度的转换：

编写一个头文件，包含下面两个函数：

double celsius\_to\_fah(double cel)    //摄氏温度到华氏温度

double fahrenheit\_to\_cels(double fah) //华氏温度到摄氏温度

实现头文件，并编写测试程序，调用函数显示如下结果：

Celsius    Fahrenheit   |   Fahrenheit       Celsius

40.0       105.0        |   120.0            48.89

39.0       102.0        |   110.0            43.33

……       ……        |   ……             ……

31.0        87.8        |   30.0             -1.11

（测试程序为主模块，即main( )函数所在的CPP文件，头文件mytemperature.h只有函数声明；函数定义写在另一CPP文件mytemperature.cpp）

4、创建名为mytriangle.h的头文件，包括：

bool is\_valid(double side1,double side2,double side3)

double\_area(double side1,double side2, double side3)

面积=sqrt(s(s-side1)(s-side2)(s-side3))

其中s=(side1+side2+side3)/2

写测试程序：读取三角形三边长，如输入合法，计算面积，否则输出错误信息。

（测试程序为主模块，即main( )函数所在的CPP文件，头文件mytriangle.h只有函数声明；函数定义写在另一CPP文件mytriangle.cpp）

**3与4选一个完成**

5、猴子吃桃：猴子第一天摘若干桃子，当即吃了一半，还不过瘾，又吃了一个。第二天又将剩下的桃子吃掉一半，又多吃一个，以后每天如此，到第10天，发现只剩最后一个桃子，问，第一天猴子共摘多少桃子（用递归实现）。

**三、实验思考题**

1. 本实验中函数中返回的值为什么与函数类型一致？

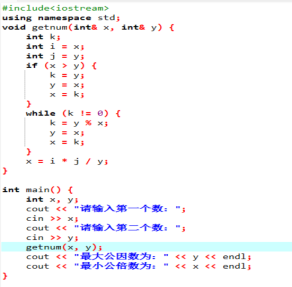
函数的返回值与函数的类型一致是因为函数类型是由返回值的类型来决定的。

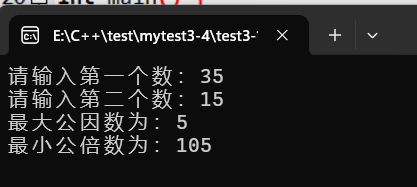
1. 本实验中主函数调用函数时采用的是何种传递方式？

第一题采用地址传递，二、三题为值传递

**四、算法分析，程序结果**

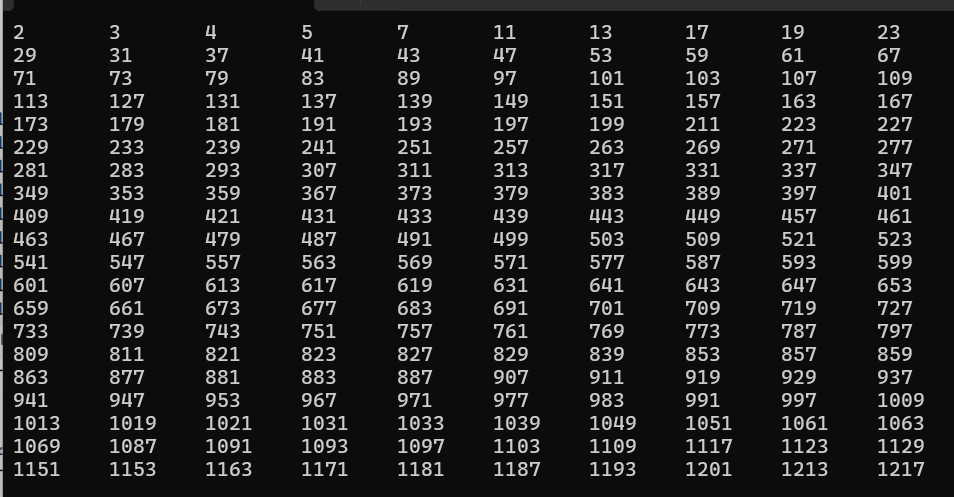
1.通过辗转相除法来求两个数的最大公约数，然后两个数相乘再除以最大公约数可得到最小公倍数。主要在与使用引用作为参数，以改变main函数中实参的值，同时引用作为参数也更加安全。

程序：

结果：

2.通过简单的遍历然后if判断即可实现判断一个数是否为素数，然后采用while循环，同时对找到的素数的个数进行计数，共有200个时才停止。

程序：

结果：

3.在头文件中进行两个函数的声明，并添加#include<iostream>与using namespace std;以便在mytemperature.cpp中可以使用标准函数，同时注意在主程序的上方进行文件的引入。

结果：

程序：

要理清楚其中的逻辑，然后写出递归函数，函数有一个参数表示第几天，然后返回那天的桃子数，通过递归调用以及边界条件的判断即可实现（第十天时还剩一个桃子）

程序：

结果：

**五、遇到的问题与解决方法**

对用引用作为函数的参数不太熟悉，后面网上查阅相关资料才了解了具体的程序要如何编写。开始时对第5题的递归程序编写没有思路，后面想通以天为参数，桃子数为返回值，问题就迎刃而解了。

**六、体会**

合理的运用递归函数能让复杂的问题通过简单的方法得到解决，不过递归函数的效率偏低，能通过递推表达式求解就尽量采用递推表达式。引用的优点是通过给同一块内存区域其别名的方式在函数内部实现参数的修改，同时节省空间，不过如果不期望修改参数时，最好采用形参。

**实验四 数组与指针**

【**实验目的**】

1、进一步加深对数组的理解，掌握数组的定义方法；

2、掌握数组的处理方法、数组作为函数参数的使用方法，以及搜索与排序的应用。

3、掌握指针的概念、指针变量定义格式以及指针的运算；

4、掌握指针与数组、函数的关系；

5、理解内存动态分配的含义、熟练掌握内存动态分配方法；

6、掌握递归函数的定义方法。

【实验内容与步骤】

**（一）数组**

1、打印不同的数：

编写一个程序，读入10个数，输出其中不同的数（即如果一个数出现多次，只打印一次）。

提示：读入的数如果是一个新的值，则将其存入一个数组。否则，将其丢弃。输入完毕后，数组中保存的就是不同的数。

下面是一个运行样例：

Enter

Enter ten numbers: 1 2 3 2 1 6 3 4 5 2

The distinct numbers are: 1 2 3 6 4 5

2、起泡排序：

利用起泡排序算法编写一个排序函数。起泡排序算法分若干趟对数组进行处理。每趟处理中，对相邻元素进行比较。若为降序，则交换；否则，保持原顺序。此技术被称为起泡排序（bubble sort）或下沉排序（sinking sort），因为较小的值逐渐地“冒泡”到上部，而较大值逐渐下沉到底部。

算法可描述如下：

bool changed = true;

do

{

changed = false;

for (int j = 0; j < listSize – 1; j++)

if (list[j] > list[j+1])

{

swap list[j] with list[j+1];

changed = true;

}

} while (changed);

很明显，循环结束后，列表变为升序。容易证明do循环最多执行listSize – 1次。

编写测试程序，读入一个含有10个双精度数字的数组，调用函数并显示排列后的数字。

3、游戏：存物柜问题：

一个学校有100个存物柜，100个学生。开学第一天所有存物柜都是关闭的。第一个学生（记为S1）来到学校后，打开所有的存物柜。第二个学生S2，从第二个存物柜（记为L2）开始，每隔两个存物柜，将它们关闭。第三个学生S3从第三个存物柜L3开始，每隔三个，将它们的状态改变（开着的关上，关着的打开）。学生S4，从L4开始，每隔四个改变它们的状态。学生S5，从L5开始，每隔五个改变状态。依此类推，直至学生S100改变L100的状态。

当所有学生完成这个过程，那些存物柜是开着的？编写一个程序求解此问题，显示所有开着的柜子号码，号码之间用一个空格隔开。

提示：使用一个100个布尔型元素的数组，每个元素代表存物柜是开（true）或关（false）。最初所有的储物柜都是关闭的。

4、合并两个排列好的数组：

编写如下函数，合并两个排列好的数组，形成一个新的排列好的数组。

void merge(const int list1[], int size1, const int list2[], int size2, int list3[])

使用size1+size2次比较实现函数。编写测试程序，提示用户输入两个排列好的数组，并显示合并以后的数组。下面是一个运行样例。注意，输入数据的第一个数字是数组的元素数，而不是数组的一部分。假定数组大小不超过80。

Enter

Enter

Enter list1: 5 1 5 16 61 111

Enter list1: 4 2 4 5 6

The merged list is 1 2 4 5 5 6 16 61 111

5、检验子串：

编写如下函数，检验C字符串s1是否是C字符串s2的子串。如果匹配，返回s1在s2中的下标，否则返回–1。

int indexOf(const char s1[], const char s2[])

编写测试程序，读入两个C字符串，检验C字符串s1是否是C字符串s2的子串。下面是程序的运行样例：

Enter

Enter

Enter

Enter the first string: welcome

Enter the second string: We welcome you!

indexOf(“welcome”, “We welcome you!”) is 3

Enter

Enter the first string: welcome

Enter the second string: We invite you!

indexOf(“welcome”, “We invite you!”) is –1

6、字符串中每个字母出现的次数：

请使用如下函数头编写函数，数出字符串中每个字母出现的次数。

void count(const char s[], int counts[])

counts是一个有26个元素的整数数组。const[0]，const[1]，…，const[25]分别记录a，b，…，z出现的次数。字母不分大小写，例如字母A和字母a都被看作a。

编写测试程序，读入字符串并调用count函数，显示非零的次数。下面是程序的一个运行样例：

Enter

Enter a string: Welcome to New York!

c: 1 times

e: 3 times

k: 1 times

l: 1 times

m: 1 times

n: 1 times

o: 3 times

r: 1 times

t: 1 times

w: 2 times

y: 1 times

**（二）指针**

1、上机验证下列程序的运行结果（有错误的话自己补充完善）

(1) void main()

｛

int i,j,\*pi,\*pj; //此处的\*表示定义指针变量，而非间接运算符

pi=&i;

pj=&j;

i=5;j=7;

cout<<i<<’\t’<<j<<’\t’<<pi<<’\t’<<pj;

cout<<&i<<’\t’<<\*&i<<’\t’<<&j<<’\t’<<\*&j;

}

运行结果：

上述结果中，pi与&i,pj与&j是地址值，随编译程序而变化，不确定。

(2) int main() //C语言程序，要了解

{

int a[]={1,2,3};

int \*p,i;

p=a; //将数组a首地址送给p

for (i=0;i<3;i++)

printf("%d,%d,%d,%d\n",a[i],p[i],\*(p+i),\*(a+i)); //与cout功能差不多

}

运行结果：

1,1,1,1

2,2,2,2

3,3,3,3

通过这两道题目，希望学生掌握数组元素与指向数组的指针的不同。

a[i]表示数组中下标为i的元素。

a[i]←p[i]←\*(p+i)←\*(a+i)

a是数组名，表示数组首地址，(p+i)表示数组中第i个元素的地址，\*(p+i) 相当于a[i]。

(3)通过如下的问题理解递归函数的定义与调用（递归未讲，可以后做）

//#include “stdio.h”

void f(char \*st,int i)

{

st[i]=’\0’;

cout<<st; // printf(“%s\n”,st);

if (i>1) f(st,i-1);

}

void main()

{

char st[]=”abcd”;

f(st,4);

}

补充完整，运行时输出为\_\_\_\_\_\_\_\_

(4)下面程序的主函数中能保证p[0]输出1，p[1]输出2吗？如何修改以保证之（提示：在函数f中使用new生成动态数组；在main中用delete释放。）

#include<iostream>

using namespace std;

int \*f()

{

int list[]={1,2,3,4};

return list;

}

void main()

{

int \*p=f();

cout<<p[0]<<endl;

cout<<p[1]<<endl;

}

2、程序设计

(1)编写函数检查字符串s1是否为字符串s2的子串，若是，返回第一次匹配的下标，否则返回-1。在主程序中输入字符串s1与s2，调用函数实现。

函数原型：int indexof(const char \*s1,const char \*s2);

(2)编写一个函数将以字符串形式表示的一个16进制数转换为10进制数，并在主函数中测试。函数原型 int parseHex(const char \*const hexString);

如：调用函数 parseHex(“A5”);返回165

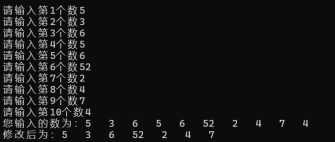
1. 主程序中建立一动态数组（使用new），数组元素及元素个数由键盘输入，动态调试观察指针及指针指向的内容；设计一个函数对数组由小到大排序；主程序中用指针方式输出数组元素；最后释放数组内存（delete）。

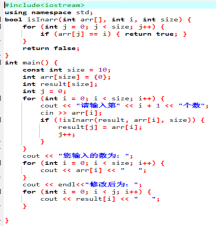
【完成实验报告】

**实验报告只要求写程序设计部分**

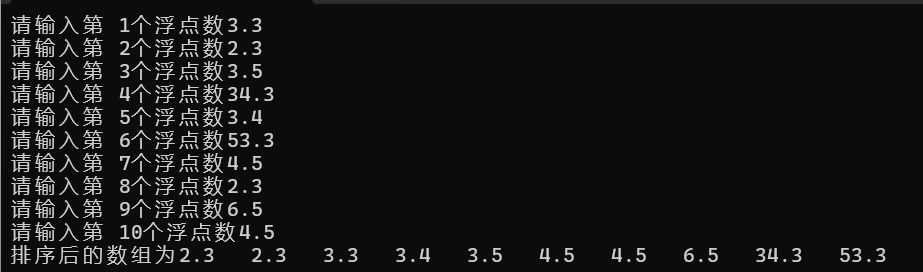
**三、算法分析，程序结果**

**1.通过for循环遍历数组来实现数据的录入，然后初始化一个result数组，长度为10，数值为0，用来存放结果。录入数据的同时进行判断，判断该数是否在结果数组中，如果不在，就添加进去，并记录当前添加进数组的数字个数，以方便后面结果的打印。**

**结果：**

程序：

2.按照题中的思路编写一个气泡排序的程序，传入一个数组以及其长度，然后实现数组的升序。

结果：

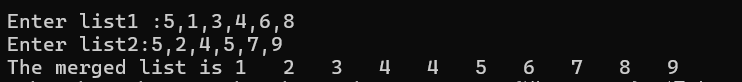
程序：

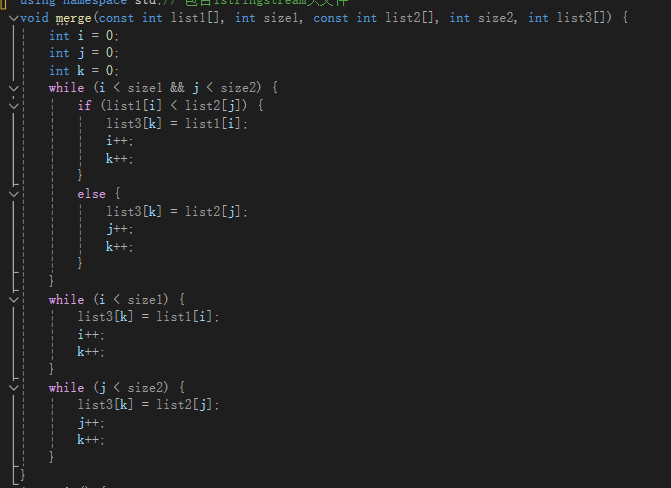
3.编写一个change函数，参数为一个数组，以及总人数size，然后在函数内部，通过for循环，每次循环对应一个同学开关柜子，通过对数组的值取反来进行开关的操作，可以很简便的实现。

结果：

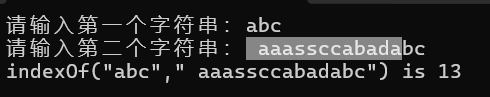
程序：

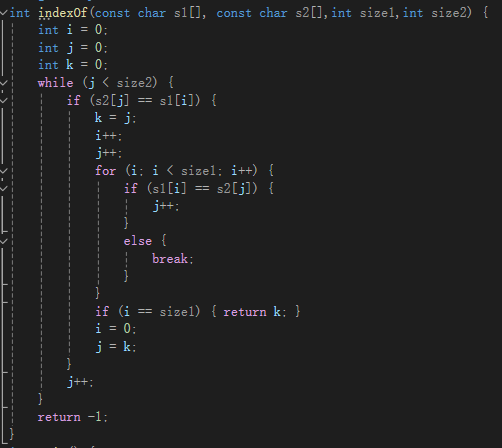
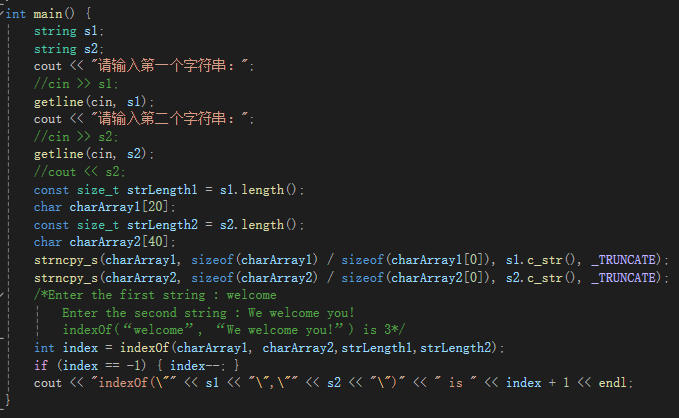
4.要合并两个排列好的数组，可以采用两个变量，分别指向数组的初识索引，然后对其指向的值进行比较，选取最小的那个进行赋值给目标数组，然后把该索引++，指向下一个位置，用while循环来实现，当其中一个索引指向其数组的末尾时退出，然后在将另一个索引指向的数组中未赋值的数全部复制到目标数组中，即可实现。主要难点在于如何输入一个字符串并把它按空格分隔为一系列数字，然后上网查阅，可以通过getline函数来将字符串按指定分割符分割，不过尝试了很多次，都无法实现按空格分割，于是就只能按逗号来分。

结果：

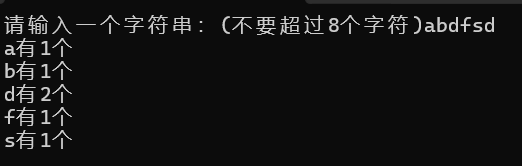
程序：

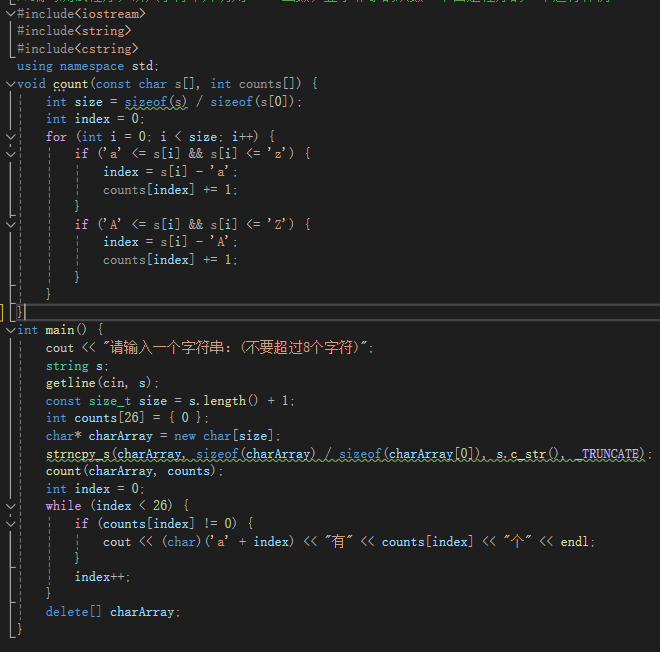
5.检验字串的主要思路就是先设置两个变量，分别指向两个字符数组的第一个索引，然后设置一个while循环，在主字符串被遍历完后退出，然后在while循环中进行判断，如果当前两个索引指向的值相同，就进行for循环，依次将子串中的每个字符与主串中的字符比对，如果有一个匹配不上，就退出循环，然后复原索引，重新匹配，全部匹配上则直接return循环开始前的索引。主要难点还是在于如何将一个字符串分割为字符数组，在网上查阅学习到可以通过strncpy\_s来实现。

结果：

程序：、

6.定义一个函数用来遍历字符数组并将结果储存到目标数组中，要计算每个字符出现的次数，可以通过将每个数组的字符进行-’a’或者-’A’的操作，得到他们是第几个字母，然后将其对应在目标数组中的索引的值++即可。后续打印就遍历目标数组，若该元素不为0，就打印该索引下对应的字母的数量。难点还是在于将字符串分割为字符数组，而且cin录入字符串存在不读取空格以及遇到特殊字符后就停止录入的问题，如377 333就只能读到377，导致数据缺失，后面采用getline(cin,变量名)的方式可以解决。

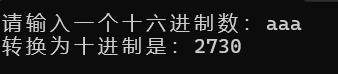
结果：

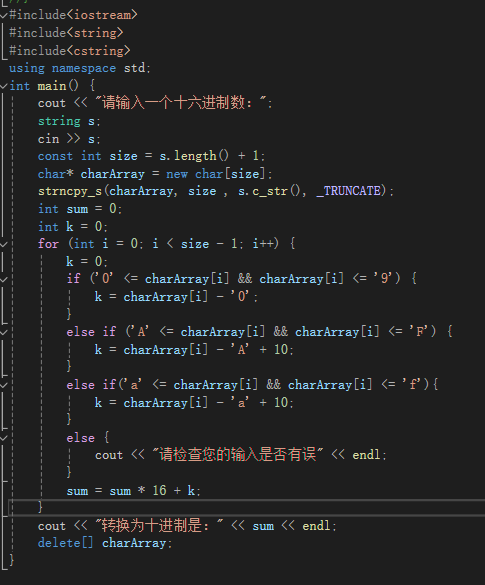
程序：

程序设计：

4\_2-2:

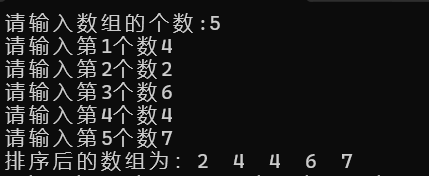
十六进制转换为十进制，先将十六进制分割为字符数组，然后将每次字符转变为对应十进制的数，在将其进行运算即可。

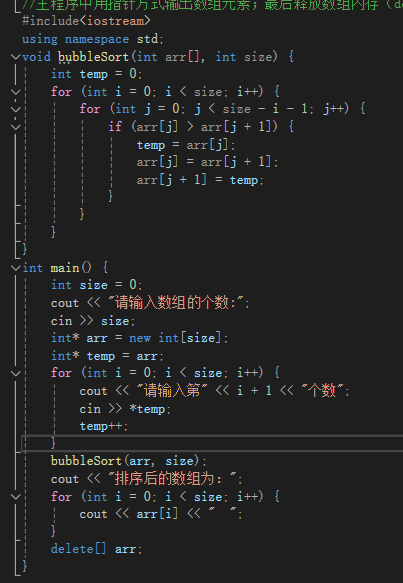
结果：

程序：

4\_2-3:

使用int\* arr = new int[size];的方式生成一个动态数组，然后定义一个指针指向该数组，并进行for循环遍历，来对数组进行赋值，最后打印并释放内存。排序采用前面设计的冒泡排序即可。

结果：

程序：

**四、遇到的问题与解决方法**

**主要的问题有如何将字符串全部分割为字符数组，以及按指定分割符分割为字符数组，还有就是如何键盘录入一个完成的字符串，并去掉换行。还有就是如何生成一个变长数组或者动态数组以及如何通过指针来对数组进行赋值等。字符串按分隔符分割我采用的是getline方法，传入分割出的子字符串的变量名，通过while循环来操作。分割为字符数组我是用的strncpy\_s，不过这个函数存在一个问题就是它在我的电脑上运行是只能接受最多8位字符的分割，多了则会报错，目前没有找到好的解决办法。动态数组以及指针的赋值比较简单，每次循环对指针++就能直接指向下一个需要填入的区域的地址。**

**五、体会**

动态数组可以节省空间，不过使用完需要手动进行释放。