实验七、数组

班级：机设（3）班 学号：115040100304 姓名：章文瑾

1. 目的要求
2. 熟练掌握一维数组、二维数组的定义、赋值、输入、输出方法。
3. 熟练掌握一维数组以及一维数组元素的储存方式。
4. 熟练掌握一维数组和二维数组的使用以及在函数中的传递。
5. 熟练掌握字符数组和字符串函数的使用。
6. 实验步骤与内容

一维数组和二维数组的定义以及数组元素的引用、字符数组的定义以及存储方式、数组名和数组元素作为函数参数以及数组有关的算法。

1. 实验小结
2. 定义整型数组的时候，使用如下语句“int a[n]”，这里要注意，“int”表示该数组的类型为整型，方括号里的n代表的是该数组的长度，也就是占有几个存储空间。这里的“a[n]”与后面输出时的“a[m]”含义是不一样的。后面语句中的“a[m]”代表的是该数组中下标为n的那个数。因为下标从“0”开始，到“长度n减去1”为止，所以的下标“m”最大值为“n-1”。你想输出数组中第三个数字，就要写“cout<<a[2];”而不是“a[3]”。
3. 在寻找数组中最大值或者最小值的时候，常先随意指定一个最大值或者最小值（通常指定首元素），然后再用该数组的每一个元素与之比较，如果比它大（或小），那么就用就这个元素来替代这个最大或最小值。经过一轮比较下来，最大或者最小的元素也就显现出来了。有时候我们不仅希望找到最大（最小）数的值。还希望知道它的位置，那么我们可以先写“int k；”用这个k来记录元素的下标。每一次交换，就记录下新的最大元素的下标。例如“if(max<a[i]) {max=a[i];k=i;}”。要比较该数组的每一个元素，常使用“for(i=0;i<n;i++)”语句对a[i]进行遍历。
4. 在该实验的上机题3中，要用程序实现插入一个数到一个降序排列的数组之中，我们可以先把这个数放在该数组的最后一位，然后与之前面的元素进行比较，如果该元素比前面的元素大，那么就交换它们的值。这种处理的思想有些类似于排序方法中的“冒泡法”，但是由于该数组其他元素已经是排好序的了，所以只需要将新加入的元素排序即可。也就是将该元素”冒泡“出去即可，而不要对该数组整体”冒泡“很多次。所以这里我们可以使用一个”break；“语句，当前面的元素比它大的时候，就无需再比较下去了，就可以直接结束该比较的阶段了，节省运算的步骤。
5. 定义二维数组时候，大致做法和前面一维数组处理一致。如下“int a[n][m];”这里的n是行数，m是列数。行数值可以缺失不写，但列数值则必须指定，不然则会出现错误。对二维数组的遍历使用两次“for”语句，则可以对该数组的行和列分别进行遍历。
6. 在上机题5之中，程序定义了一个对数组进行处理函数。然后在后面的程序使用该函数对某数b[]组进行处理之后，再一次输出b[]的值时，就是处理之后的数值了。这是函数形参与实参之间的对应关系。在数组函数的运用之中需要特别注意。
7. 在字符型数组的定义之时，使用“char”语句，例如：“char a[6]=”basic”;”这里注意，定义该数组长度时要比字符的个数至少多1个。因为系统在存储字符串的时候会在末尾自动追加一个字符‘\0’，如“basic”有5个字符，实际存储时是“basic\0”。定义长度为6，就是因为考虑到末尾的‘\0’，如果在这里把数组长度定义成5，那么在编译程序是会出现“array bounds overflow”的错误，意思是数组的边界溢出。
8. 在输入字符型数组时，不需要像数组一样逐个输入，而是可以整体输入，直接使用“cin>>a[]”即可。但是这里需要注意，这里输入的时候，字符中不能有空格。如果要输入有空格的，可以使用“gets()”语句。输出则可以使用“puts()”语句。这样输出时遇到“\0”自动停止，不需要写“for(i=0;str[i]!=0;i++)”来输出数组。
9. 在甄别字符类型时，可以用“if(a[i]>’a’&&a[i]<=’z’||a[i]>=’A’&&a[i]<=’Z’)”来判断是否是字母字符，用“if(a[i]>=’0’&&a[i]<=’9’)”来判断是否是数字字符。
10. 上机题9需要计算一个二维数组的周边元素之和，我们可以先对一个二维数组进行遍历，然后选择出周边元素进行相加求和。因为周边元素的特点为“i==0 || j==0 || i==N-1 || j==M-1”。所以将符合这些特点的元素相加之后，便得到需要的结果了。
11. 上机题10需要平移数组中字符串的某一小段内容，实现该功能的部分程序如下：

for( int i = 0; i < m; i++)

{

c = s[0];

j = 1;

while( s[j] != '\0')

{

s[j-1]=s[j];

j++;

}

s[j-1]=c;

}

因为要将字符串整体的往前平移，第一个字符会被后面的字符掩盖掉，所以要在执行替换之前先保存第一个字符，程序中利用一个“c=s[0]”来实现。最后再将这个c值赋给最后一个字符。这样的一次便将第一个字符移到了最后面形成了新的排列。需要平移多少长度段的字符，就这样循环多少次即可。

1. 在定义数组的时候，我们常在程序的开头写“#define N 10”这样在之后的数组定义之中，可以直接写“ char a[N]”，这样当我们想修改数组长度时，可以直接在define里面修改，免去了逐个修改的麻烦。在上机题11中，我们需要校验一个身份证号码，首先我们需要校验前17位是否全是数字，可以通过如下语句实现：

While (s[i]!=’\0’&&i<17)

{

If(s[i]<’0’||s[i]>’9’)

{

F=0;

Break;

}

}

这里使用了一个break语句，因为一旦前17位有一个不是数字，就不需要再进行校验下去了。然后再对这17位数字进行加权求和。这里我们还要注意另一种情况，就是位数多了，既第九位不是“\0”,所以用如下语句筛选掉：

“if（s[18]!=’\0’） f=0”。然后再对前17位数加权求和，看是否等于第18位校验码即可。