

实验七：数组与排序(2 学时)

一、实验方式：上机

二、实验目的：

- 1、熟悉经典的排序方法—冒泡排序。
- 2、熟悉选择排序的基础—交换排序。

三、实验内容及其步骤：

1、在实际开发中，有很多场景需要我们将数组元素按照从大到小（或者从小到大）的顺序排列，这样在查阅数据时会更加直观。例如，一个保存了商品单价的数组，排序后更容易看出它们的性价比。对数组元素进行排序的方法有很多种，比如冒泡排序、归并排序、选择排序、插入排序、快速排序等，其中最经典最需要掌握的是「冒泡排序」。整个排序过程就好像气泡不断从水里冒出来，最大的先出来，次大的第二出来，最小的最后出来，所以将这种排序方式称为冒泡排序。

冒泡排序的整体思想是这样的：从数组头部开始，不断比较相邻的两个元素的大小，让较大的元素逐渐往后移动（交换两个元素的值），直到数组的末尾。经过第一轮的比较，就可以找到最大的元素，并将它移动到最后一个位置。第一轮结束后，继续第二轮。仍然从数组头部开始比较，让较大的元素逐渐往后移动，直到数组的倒数第二个元素为止。经过第二轮的比较，就可以找到次大的元素，并将它放到倒数第二个位置。以此类推，进行 $n-1$ （ n 为数组长度）轮“冒泡”后，就可以将所有的元素都排列好。

编程实现数组的冒泡排序 `int num[4] = {4, 5, 2, 1}; n=4`

第一轮($i=0$)，比较两个数字 (4, 5)，不交换位置；比较 (5, 2)，交换位置 (2, 5)；比较 (5, 1)，交换位置 (1, 5)；此时，最大元素 5 被移动到了最后一位，但其他元素还是乱的。此时，`int num[4] = {4, 2, 1, 5};` 比较进行了 3 次 ($j=n-1-i$)。

第二轮($i=1$)，再次从头比较。比较 (4, 2)，交换 (2, 4)；比较 (4, 1)，交换 (1, 4)；4和5就不用比较了，因为第一轮已经确立了 5 最大的地位。此时，`int num[4] = {2, 1, 4, 5};` 比较 2 次 ($j=n-1-i$)。

第三轮($i=2$)，再次从头比较。比较 (2, 1)，交换 (1, 2)；2和4就不用比较了，因为第二轮已经确立了4的次大地位。此时，`int num[4] = {1, 2, 4, 5};` 比较 1 次 ($j=n-1-i$)。

前三轮确定了3个数的位置，因为这里一共有四个数，所以剩下的一个数不需再比较，就是最小的。所以 n 个数，只需排 $n-1$ 轮即可。

编程时候注意，每一轮确定一个正确位置的数，每一轮还存在一个比较次数的问题。而数组的下标是从0开始。

```

#include<stdio.h>
int main(void)
{
    int num[4] = { 4, 5, 2, 1 };
    int i, j, temp;
    // n-1 轮排序
    for (i = 0; i < 4-1; i++)
    {
        // 每一轮比较前 n-1-i 个，已经排好的最后 i 个不用比较。
        for (j = 0; j < 4 - 1 - i; j++)
        {
            if (num[j] > num[j+1])
            {
                temp = num[j];
                num[j] = num[j+1];
                num[j+1] = temp;
            }
        }
    }
    //输出排序后的数组
    for (j = 0; j < 4; j++)
    {
        printf("%d ", num[j]); //留有空格
    }
    return 0;
}

```

2、交换排序是选择排序的基础，或者说选择排序是交换排序的升级版，本次上机只熟悉交换排序，详细实施过程见课本 P202。

编程实现对上题中数组元素 `int num[4] = {4, 5, 2, 1}` 的进行交换法从小到大排序。

```

#include<stdio.h>
int main(void)
{
    int num[4] = { 4, 5, 2, 1 };
    int i, j, temp;
    // n-1 轮排序
    for (i = 0; i < 4 - 1; i++)
    {
        for (j = i+1; j < 4 ; j++)
        {
            if (num[i] > num[j])

```

```
        {  
            temp = num[i];  
            num[i] = num[j];  
            num[j] = temp;  
        }  
    }  
}  
//输出排序后的数组  
for (i = 0; i < 4; i++)  
{  
    printf("%d ", num[i]);    //留有空格  
}  
return 0;  
}
```

3、自行编写程序，如何分别改写上述两种方法实现 `int num[4] = {4, 5, 2, 1}` 从大到小的排序。
提示：方法有很多。