C语言学习体会

C语言开课已有一个月，在这个过程的学习中也收获了许多知识，以及学习过程中的心得体会。

C语言虽有语言二字，但与我们平时的中文英文等与人交流的自然语言不同，C语言是与机器交流的高级语言，它具有以下的特点：

（1）简洁的语言

C语言包含的各种控制语句仅有9种，关键字也只有32 个，程序的编写要求不严格且以小写字母为主，对许多不必要的部分进行了精简。实际上，语句构成与硬件有关联的较少，且C语言本身不提供与硬件相关的输入输出、文件管理等功能，如需此类功能，需要通过配合编译系统所支持的各类库进行编程，故c语言拥有非常简洁的编译系统

（2）具有结构化的控制语句

C语言是一种结构化的语言，提供的控制语句具有结构化特征，如for语句、if⋯else语句和switch语句等。可以用于实现函数的逻辑控制，方便面向过程的程序设计。

（3）丰富的数据类型

C语言包含的数据类型广泛，不仅包含有传统的字符型、整型、浮点型、数组类型等数据类型，还具有其他编程语言所不具备的数据类型，其中以指针类型数据使用最为灵活，可以通过编程对各种数据结构进行计算。

（4）丰富的运算符

C语言包含34个运算符，它将赋值、括号等均是作运算符来操作，使C程序的表达式类型和运算符类型均非常丰富。

（5）可对物理地址进行直接操作

C语言允许对硬件内存地址进行直接读写，以此可以实现汇编语言的主要功能，并可直接操作硬件。C语言不但具备高级语言所具有的良好特性，又包含了许多低级语言的优势，故在系统软件编程领域有着广泛的应用。

（6）代码具有较好的可移植性

C语言是面向过程的编程语言，用户只需要关注所被解决问题的本身，而不需要花费过多的精力去了解相关硬件，且针对不同的硬件环境，在用C语言实现相同功能时的代码基本一致，不需或仅需进行少量改动便可完成移植，这就意味着，对于一台计算机编写的C程序可以在另一台计算机上轻松地运行，从而极大的减少了程序移植的工作强度。

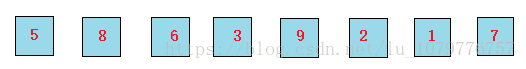
（7）可生成高质量、目标代码执行效率高的程序

与其他高级语言相比，C语言可以生成高质量和高效率的目标代码，故通常应用于对代码质量和执行效率要求较高的嵌入式系统程序的编写。

上述特点注定了C语言是从未接触过编程的大学新生了解编程的最好途径，让学生通过编写程序的方式理解计算机的行为（如何存储数据，实现人机交互等）。

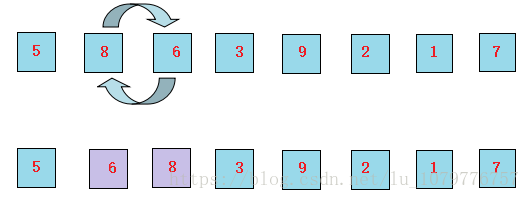
C语言的学习过程中，我同样产生了许多的心得体会。计算机是死板的，它说一不二，但在面对同一个问题时，如果我们进行深度思考，也能提供多种算法实现，下面以我学习冒泡排序法为例：

以下列八个数组成一个无序数列，现将其从小到大排序

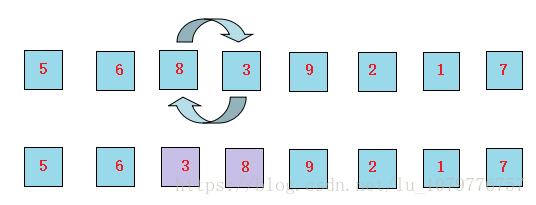


第一轮排序：

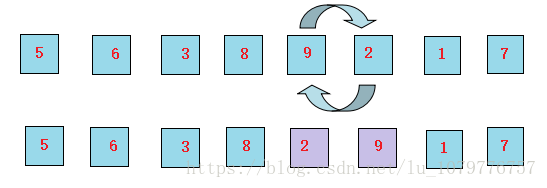
1. 首先让5，8进行交换，发现5比8小，于是元素位置不变
2. 接下来让8，6比较，8比6大，所以交换8和6的位置



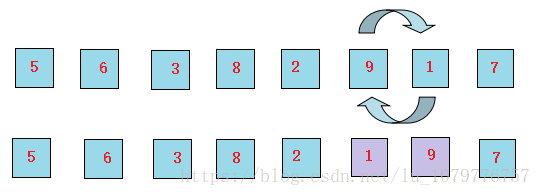
1. 继续让8和3比较，交换位置



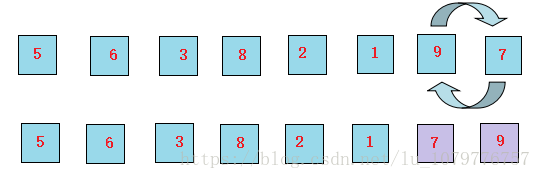
1. 让8和9比较，发现8比9小，不用交换
2. 9和2进行比较，交换位置



1. 9和1进行比较，发现9比1大，所以交换9和1的位置



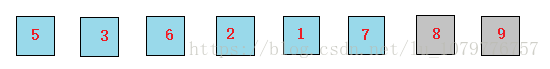
1. 最后让9和7交换位置



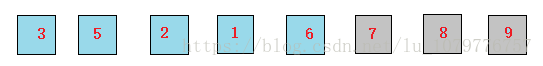
至此，就把最大的元素9移动到了最右端



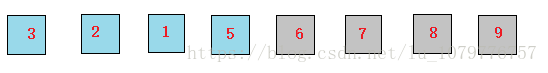
以此类推，第二轮的状态为：

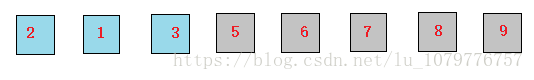


第三轮的状态为：



第四轮的状态为：

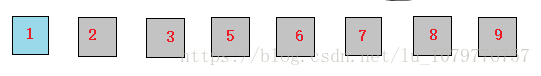
第五轮的状态为：



第六轮的状态为：（已经有序了）



第七轮的状态为：



第九轮的状态为：

这样的算法是稳定的，即进行n-1轮的排序，遍历了所有元素

代码如下：

#include<stdio.h>

int main()

{

int a[]={5,8,6,3,9,2,1,7};

int i,j,temp;

printf(“排序前:”);

for(i=0;i<7;i++)

printf(“%d “,a[i]);

for(i=0;i<7;i++)

{

for(j=0;j<7-I;j++)

if(a[j]>a[j+1])//比较相邻两个数的大小

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;//利用中间变量交换值

}

}

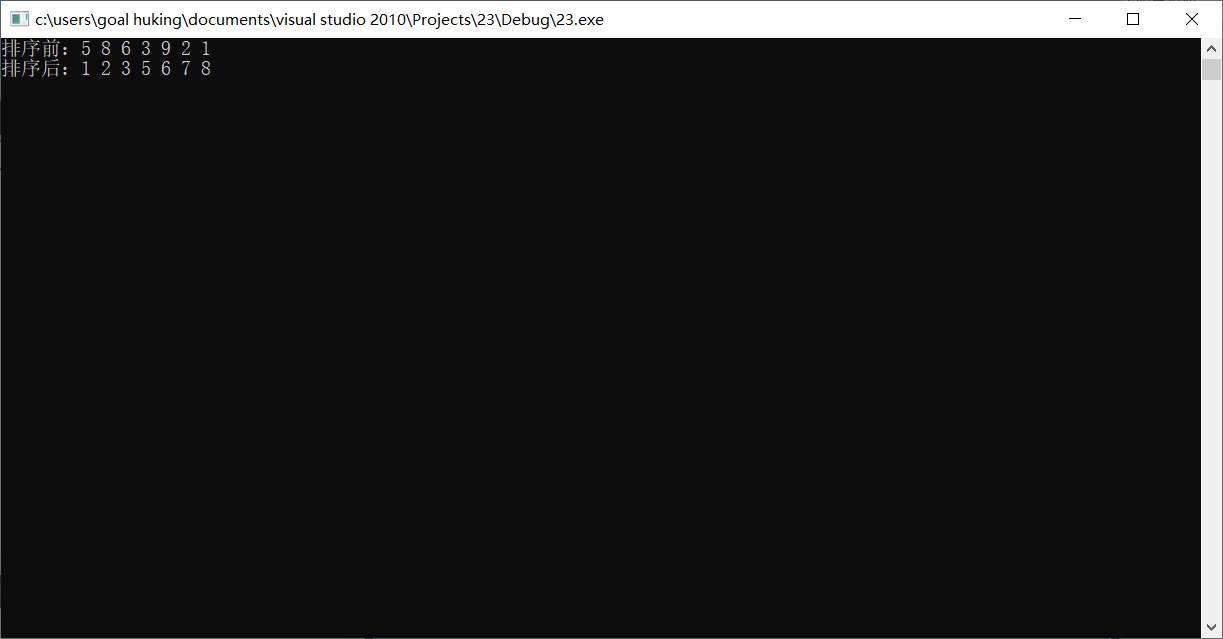
printf(“\n排序后:”);

For(i=0;i<7;i++)

printf(“%d “,a[i]);

}

输出如下



但是从上述讲解我们发现，该数列实际上在第六次排序时就已经是有序的了，因此我们可以思考如何对上面的算法进行优化。在这里我提供自己的优化算法。

#include<stdio.h>

int main()

{

int a[]={5,8,6,3,9,2,1,7};

int i,j,temp;

bool flag;

printf(“排序前:”);

for(i=0;i<7;i++)

printf(“%d “,a[i]);

for(i=0;i<7;i++)

{

flag=1;//以布尔值1为逻辑真

for(j=0;j<7-I;j++)

if(a[j]>a[j+1])//比较相邻两个数的大小

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;//利用中间变量交换值

flag=0;//以布尔值0为逻辑假

}

if(flag==1)//如果为真，则跳出循环

break;

}

printf(“\n排序后:”);

For(i=0;i<7;i++)

printf(“%d “,a[i]);

}

以布尔值1表示完全排序，0表示未完全排序。这个代码是这样运作的：每轮开始时将布尔值定义为真，随后在排序的过程中，若if语句中发生了a[i]与a[i+1]的交换，则说明还未达到完全排序，此时将布尔值定义为假，若为发生a[i]与a[i+1]的交换，则说明已完全排序，布尔值为真，随后跳出循环并输出

通过上述示例，我感到了C语言的魅力，不仅要开发算法，开发之后更是有优化算法的空间。这次对冒泡法排序的深研使我对学习C语言的兴趣更加浓厚，在接下来两个月的学习中，我将更加刻苦，钻研更多更好的算法！

20049200384

苟虎劲