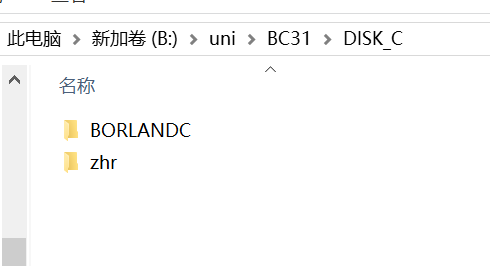
**第一次实验报告**

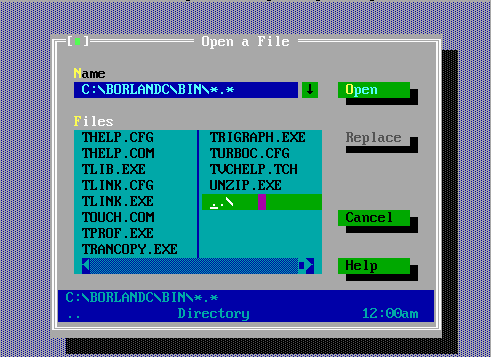
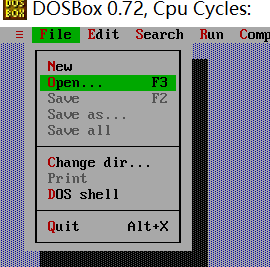
1. 建立自己的工作目录，验证自己编辑文件的存储和打开。
2. 输入一个完整的程序，熟悉集成开发环境下的菜单主要功能和功能键（编译、连接、运行、查看运行结果等等）
3. 用2个实例说明和掌握利用编译错误信息，完善源程序。
4. 至少用三个不同实例实例（顺序结构、if 语句、switch语句等）熟悉和掌握单步调试（F4、F7、F8、Alt\_F4的使用，以及断点的设置和使用）
5. 用2个实例熟悉和掌握scanf和printf函数的使用。

**一 . 建立自己的工作目录，验证自己编辑文件的存储和打开。**

**实验思路：**在DISK\_C中建立自己的文件夹，然后再BC中验证.

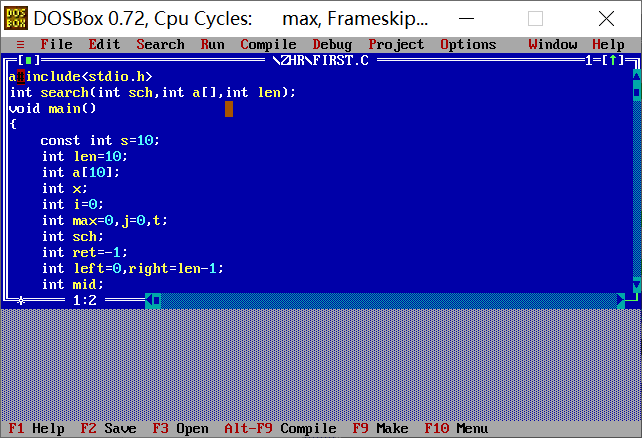
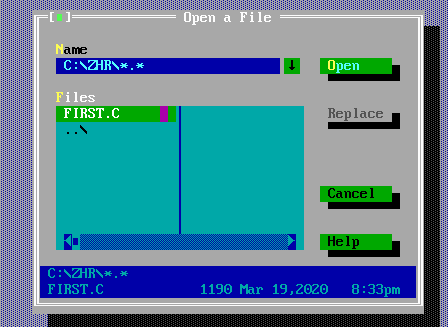
1.在DISK\_C中新建文件夹zhr，然后将提前写好的first.c放进文件夹里。

2.打开BC，打开其中的file\open。



3.将窗口内的横向光标拖至尾部，找到\..，多次点击，最终发现zhr文件夹

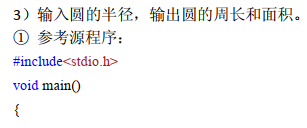
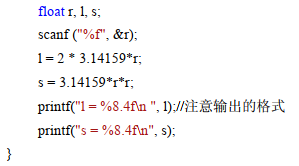
4.打开zhr文件夹，找到first.c文件，选中并打开。

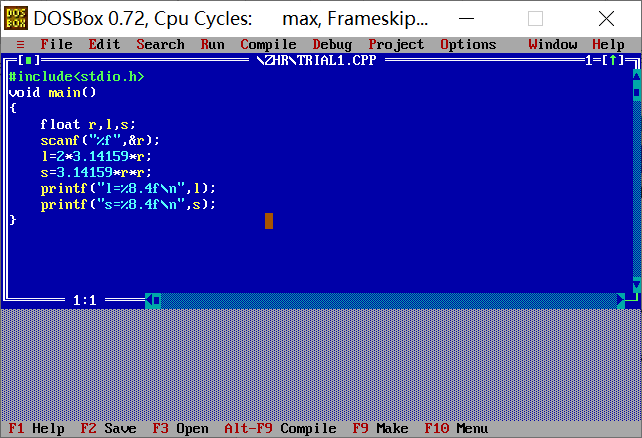


**实验感想：**为了将自己的文件和BC的文件区分开，所以在DISK\_C中额外设置自己的文件夹，值得注意的是，要先设置文件夹，然后再打开BC。

**二.输入一个完整的程序，熟悉集成开发环境下的菜单主要功能和功能键（编译、连接、运行、查看运行结果等等）**

**实验思路：先输入一个程序，然后利用CTRL F9编译连接运行，alt F5查看结果。**

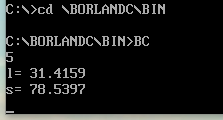
我选取了C语言实验上机指南中输入圆的半径，输出圆的周长和面积的程序。 



1.按快捷键ctrl F9进行编译，链接和运行程序，输入5



2.按快捷键alt F5打开用户界面来查看结果

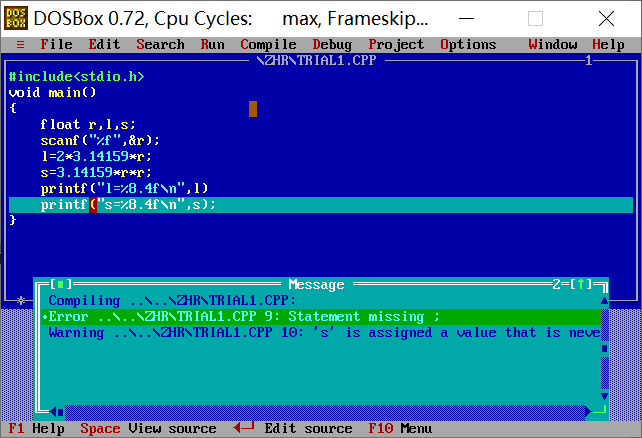


**实验感想：**编译，连接，运行是三个不同的步骤，CTRL F9可以一键解决，非常方便。

**三.用2个实例说明和掌握利用编译错误信息，完善源程序**

**实验思路：对程序进行编译，分析错误信息，修改程序，使编译通过。**

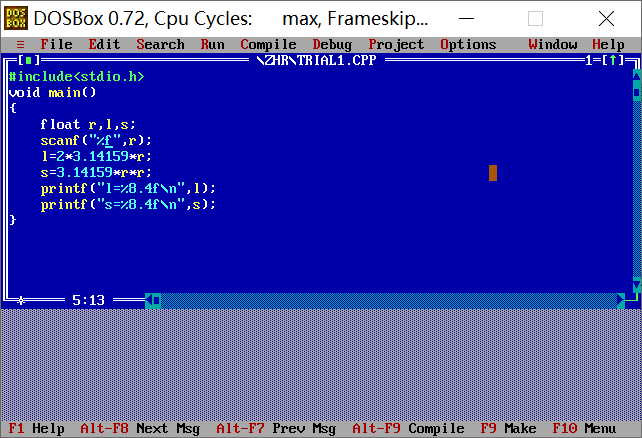
实例一：



分析：Error为第九行statement缺少”;”. 值得注意的是，实际上第九行末尾有分号，反而是第八行末尾缺少了分号（其实是电脑读完了第八行，然后开始读第九行，因为第八行的语句还少一个分号，所以电脑现在期待一个分号，但是在第九行起始并不是分号，于是error显示为第九行缺少一个分号）

解决：在第八行补上分号，编译成功。

实例二：



虽然编译成功了，但运行却出现了问题



分析：scanf函数中，浮点型数据没有被连接，再回过头去看scanf函数，发现没有用&来指向变量的地址。

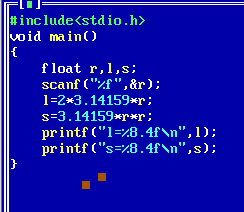
解决：在scanf中添加&，编程运行成功。

**实验感想：**有时错误信息未必能够告诉我们真实的错误，比如n行没有加分号，错误信息就会提示n+1行缺少分号，读懂电脑的意思，还需要我多多练习。

**四.至少用三个不同实例实例（顺序结构、if 语句、switch语句等）熟悉和掌握单步调试（F4、F7、F8、Alt\_F4的使用，以及断点的设置和使用）**

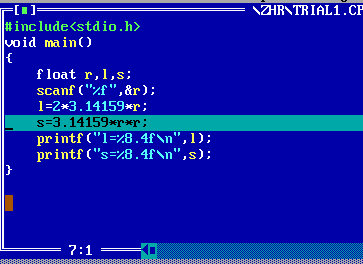
**实验思路：对于一个程序，先用F4使程序运行到指定位置，然后用alt F4观察变量值，按F7或F8让程序逐步进行。（在关键位置设置断点，方便让程序只在指定位置停止）**

实例一（顺序结构）计算圆的面积和周长(输入6)

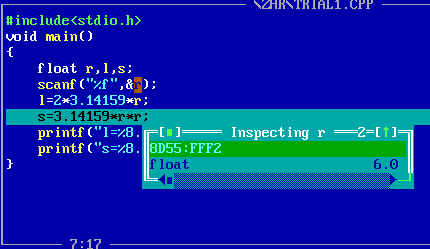


F4 运行至光标位置并暂停

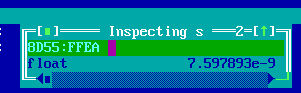
将光标放在第七行，按F4后程序运行并暂停在该行



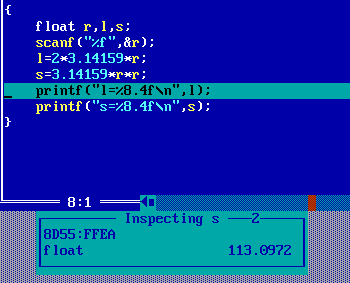
然后将光标指向r，按alt F4，显示出此时此刻r的情况



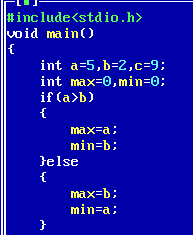
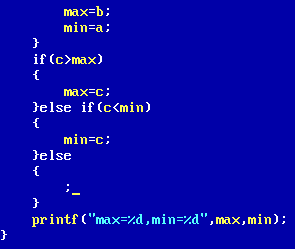
再来看看此时此刻s的值

（语句尚未执行，且s未提前赋值，所以此时s的值很奇怪）

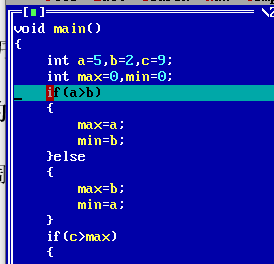
按F8执行下一条语句

可以看到，此时s的值发生改变

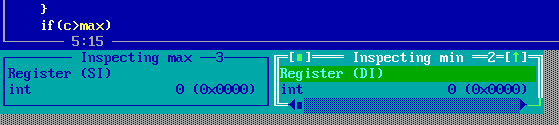
实例二（if语句）三个数字选出最大的和最小的

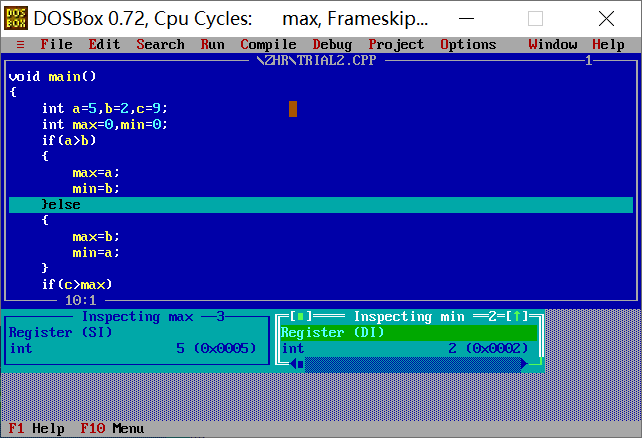
与实例一中类似，将光标放在第六行开头，按F4

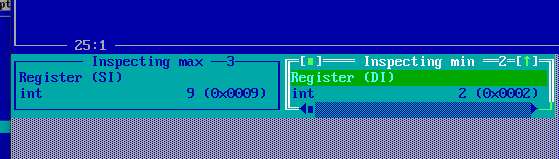


分别用alt F4观察max,min的值

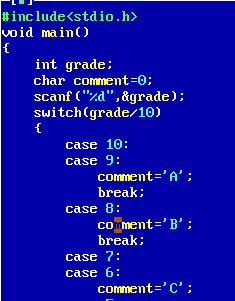
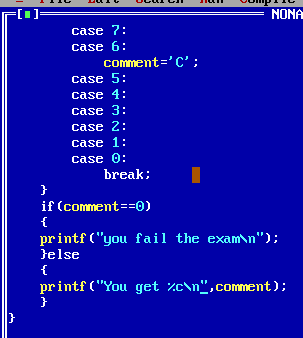


按F8逐行运行

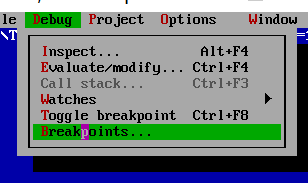
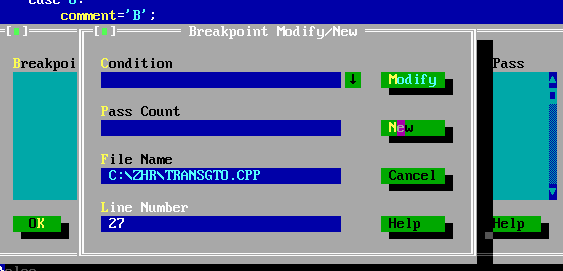
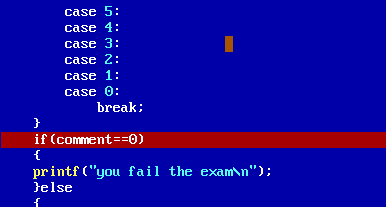




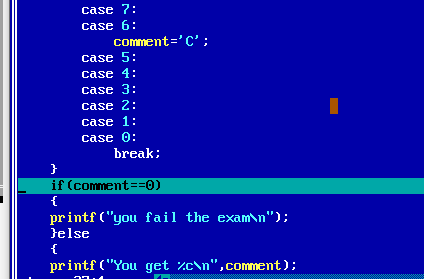
实例三（switch语句）输入一个学生的成绩，输出这门课对应的评价（90分及以上为A，89-80为B，79-60为C，59-0输出结果“you fail the exam”）

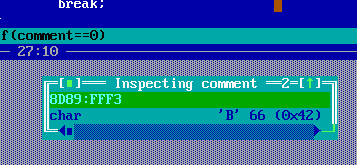
这次我们在27行设置一个断点

然后按ctrl F9编译运行，输入80，发现程序自动停在27行



然后按alt F4观察comment的值

（66是B对应的ASCII码值）

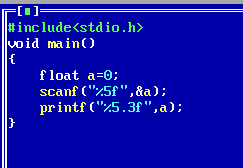
之后按F8逐行运行直至结束。

**实验感想：**注意区分F7和F8作用，其中F7可以进入函数的内部，F8则会直接跳过函数。所以，途径自定义的函数时，如果想要进入函数内部一探究竟，就用F7，其他情况只用F8就够了。

**五.用2个实例熟悉和掌握scanf和printf函数的使用。**

**实验思路：编写简单的程序，输入不同的值，探究scanf和printf的用法。**

实例1：



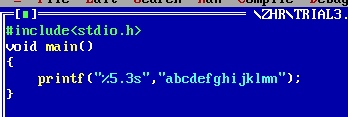
如图代码，输入数据123.123，输出结果123.100



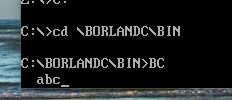
scanf中%5f,表示只输入前五位数据（小数点也算一位），printf中%5.3表示5位有效数字，其中小数点后有3位。

**实验感悟：**对于数字的输入输出%m.nf可以控制数字的有效位数以及小数点位数。

实例二：



输出结果位



输出字符串时，%5.3s表示，输出5个字符，实际只取前3个字符，左补2个空格，若程序中为%-5.3s,则输出5个字符，实际只取前3个字符，右补2个空格。

**实验感悟：**值得注意的是，比如说printf(“%5ld”,123123),实际会输出123123，多余的部分依旧会输出，这是我很容易弄混的一点。（字符串也是类似的情况）