宇创培训 WWW.SPELS.CN

360教育在线 www.eol360.com

宇创IT培训教学专用教材
360教育在线指定使用教材

QUANGUO JISUANJI DENGJI KAOSHI ERJI JIAOCHENG

全国计算机等级考试
二级教程

宇创IT培训教学研究组 编

第五分册
上机实验手册

SHANGJI SHIYAN SHOUCE

中国地质大学出版社

全国计算机等级考试二级教程

（第五分册）

上机实验手册

宇创 IT 培训教学研究组 编

中国地质大学出版社

目　录

实验一　熟悉C语言

实验要求

1.能够熟练使用TC 2.0开发环境；
2.理解C语言顺序结构；熟练掌握printf函数和scanf函数的使用方法。

特别要求

在实验中遇到困难时，切忌不要动不动就问老师，而是应该先思考。如果确实不能解决困难，先与坐在旁边的同学讨论，而无论这个同学以前是否认识。如果有同学问你问题，无论你是否能解决这个问题，都应该和他讨论。如果讨论了仍然不能解决问题，就一起问老师。讨论问题的好处在于能够深刻地理解问题、记住问题。

实验内容

一、熟悉TC开发环境

1.安装TC

将教材中的光盘放入光驱，把光盘中的TC文件夹拷贝到电脑的C盘。如果电脑没有光驱，也可以到网上下载TC(如www.spels.cn)，将下载的文件包解压到电脑C盘即可。如果电脑中已经安装TC，请在老师的指导下使用。

2.编写源程序

在TC文件夹下找到文件TC.EXE。鼠标双击该文件，打开TC。此时会出现如图1-1所示的界面。

界面上方的File、Edit等一排文字叫做菜单，可以按F10键激活或取消激活菜单。如果某一个菜单被激活，该菜单以黑底白字的方式显示，如图1-1的File菜单。

打开TC后按F10键或ESC键取消激活菜单，这时在Edit窗口中会有一个光标一闪一闪的，我们就可以在TC中编写程序了。在TC中用键盘输入下面的程序代码：

```
main( )
{
    printf("Hello,Spels!\ n");
}
```

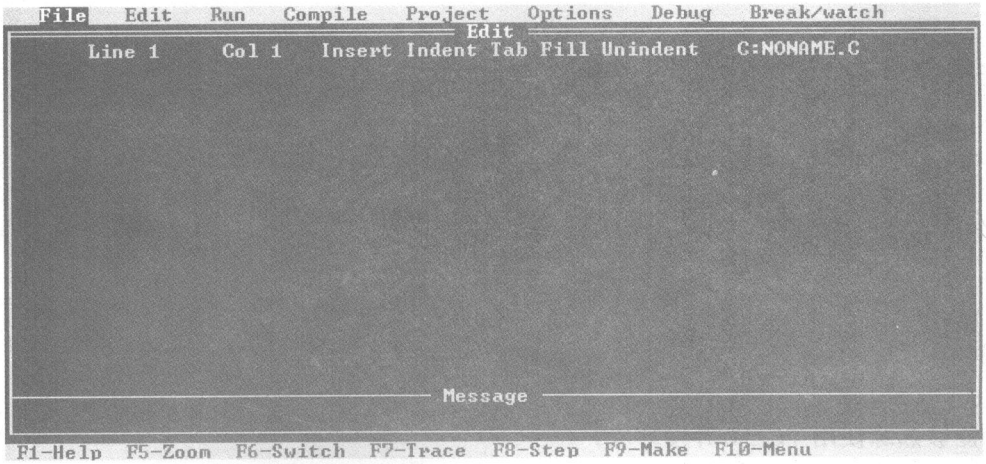

图 1-1 TC 运行界面

在输入代码时注意不要将符号输入错误。程序输入完毕后 TC 的界面如图 1-2 所示。

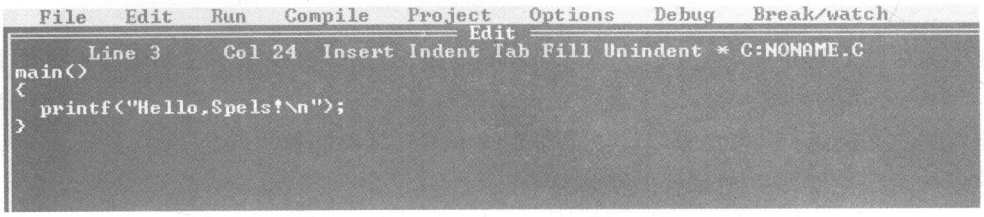

图 1-2 输入程序

3.编译、链接程序

按 F10 键激活菜单,用键盘上的方向键选择 Compile 菜单下的 Compile to OBJ 子菜单,如图 1-3 所示,然后按回车键。

注意:在 TC 中不能使用鼠标,只能使用键盘。

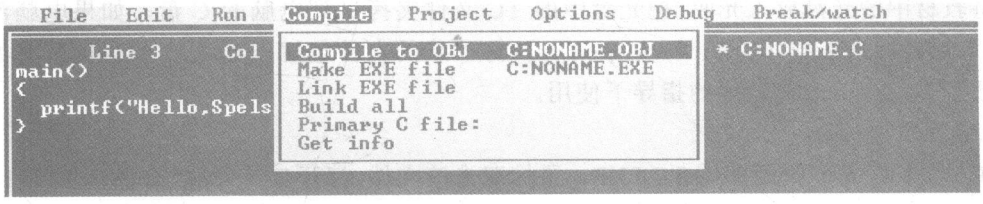

图 1-3 选择 Compile 菜单

如果程序没有任何错误,将会出现如图 1-4 所示的界面。

这个过程叫做编译,此时编译程序将我们所写的源程序编译成目标程序,生成一个以.obj 结尾的目标文件。在图 1-4 的界面中,有两行数据值得我们注意:Warnings 和 Errors。如果程序没有任何错误,这两个标记后面的数字均为 0。如果 Errors 后面的两个数据不为 0,则表示程序有语法错误;如果 Warnings 标记后面的两个数据不为 0,程序本身并没有语法错误,此时计算机认为程序可能会出现逻辑错误,因此对我们提出警告。关于排错的方法,后面会专门讲解。

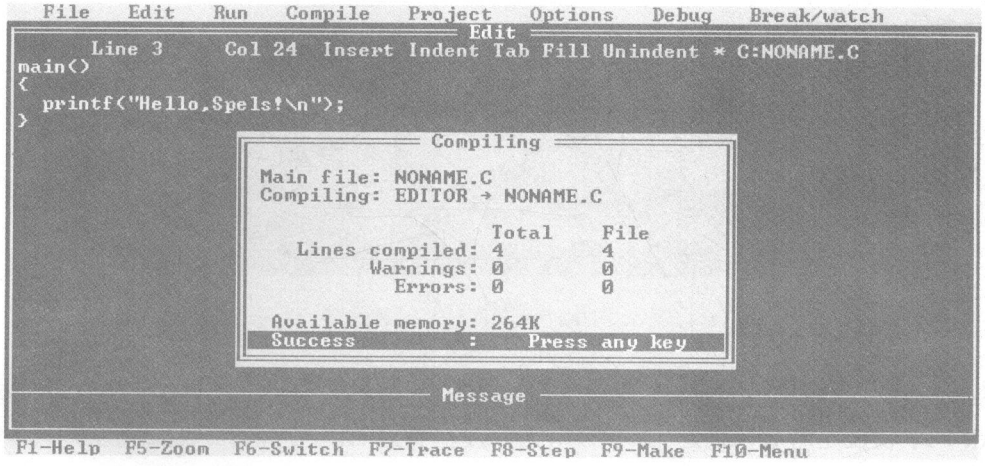

图1-4　编译程序

按任意键消除提示窗口。同样使用F10键和方向键选择Compile菜单下的Link EXE file子菜单，此时会出现一个与图1-4相似的窗口。同样，如果没有任何错误，提示窗口的Warning和Errors标记后面的数据均会为0，这个过程叫链接。链接程序将编译生成的目标文件链接，生成一个以.exe结尾的可执行文件。到目前为止，我们的第一个程序成功完成了。

打开TC文件夹，看看是否会有一个名为“NONAME.EXE”的文件。如果存在，恭喜您了，这是您所开发的第一个软件。尽管您的软件没有实现任何功能，但是，您看到了，编写软件是如此的简单，我们每一个人都能。

我们可以选择Compile菜单下的Make EXE file子菜单同时完成上面的两个步骤。

4.运行程序

选择Run菜单下的Run子菜单，可以运行我们刚才所写的软件。如果屏幕上未出现任何提示，表示程序运行成功。

为了方便起见，我们可以用组合键Ctrl+F9来运行程序。Ctrl+F9组合键的按键方法为：先用左手将Ctrl键按下；然后用右手快速按下F9键，快速松开F9键，注意按F9键的时间不要太长；最后松开Ctrl键。

在使用TC的运行程序菜单时，如果程序没有被编译、链接，TC先将程序编译、链接，然后再运行，也就是说，我们可以完全忽略前面的编译、链接程序步骤，直接使用Run菜单下的Run子菜单，或者使用快捷键Ctrl+F9。

一个程序可以在编写完成后直接使用Run菜单下的Run子菜单，或者按Ctrl+F9组合键。如果程序没有任何错误，我们只会看到屏幕上闪一下(甚至连闪一下都看不到)，不会出现任何提示；如果出现类似图1-4的界面，上面Errors标记后面的数字不为0，则表示程序有错误。

5.查看结果

运行程序时并不能看到程序的最终运行结果，我们可以使用Run菜单下的User Screen菜单查看程序的运行结果，其结果如图1-5所示。

可以看到在屏幕的左上角有一行字：Hello,Spels!，这就是我们程序输出的结果。同样，查看运行结果也有快捷组合键：Alt+F5。查看运行结果后，按任意键可以回到TC编程界面。

注意：运行程序和查看程序运行结果是完全不同的两件事。任何程序必须要运行后，才能

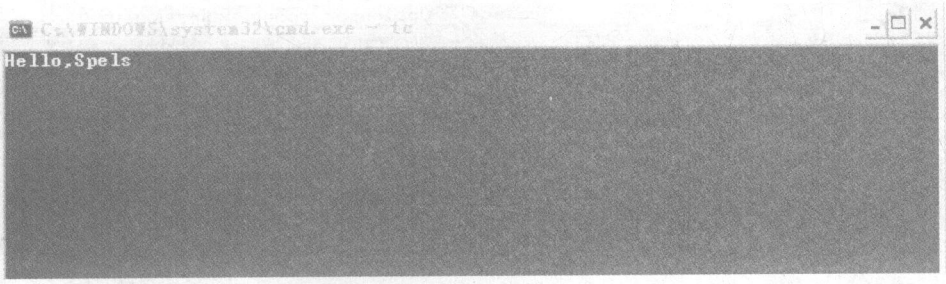

图 1-5 程序运行结果

有结果。因此在程序编写完成后必须先按 Ctrl+F9 运行程序,然后再按 Alt+F5 查看运行结果。Alt+F5 仅仅是看看前面运行的结果,无论按多少遍 Alt+F5,其界面都完全一样,不发生任何改变。但是,如果按了 Ctrl+F9,则程序又运行了一遍,此时按 Alt+F5,可以看到前后两次的界面发生了改变。

读者可以试试多按几次 Alt+F5,看看前后两次的界面有没有什么不同。然后再多按几下 Ctrl+F9,再按 Alt+F5,看看前后两次的界面又有什么不同。

总结:一个程序在编写完毕后,直接按 Ctrl+F9 运行程序,然后再按 Alt+F5 查看运行结果。

6.检查错误

任何人在编写程序的时候,都或多或少会出现一些错误。因此,学会排除错误是编写程序最基本的要求。比如在按 Ctrl+F9 后出现图 1-6 所示的界面。

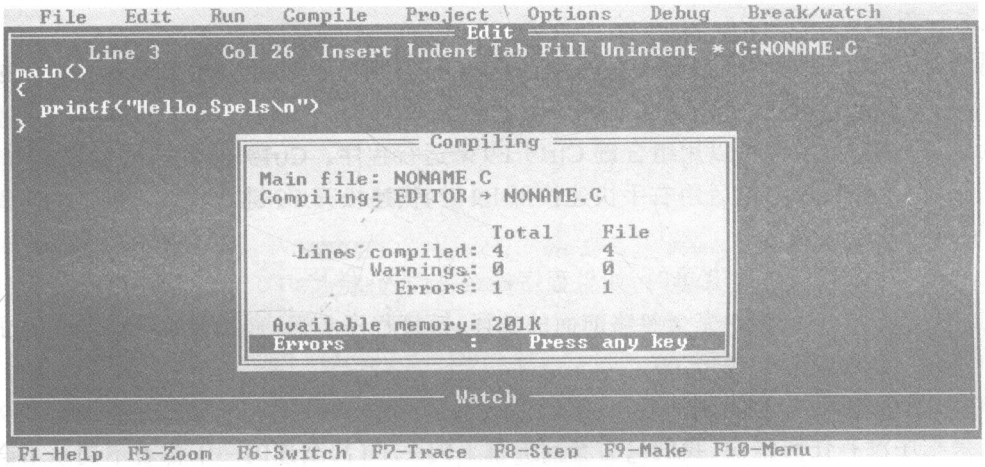

图 1-6 出错的界面

在提示窗口的 Errors 标记后面的数字为 1,表示 TC 系统发现程序存在一个语法错误,也就是说我们编写的程序有语法错误。按任意键回到编辑界面,如图 1-7 所示。

图 1-7 中有两个窗口,上面的是 Edit 窗口,即编程的编辑窗口;下面的是 Message 窗口,即信息提示窗口。可以按 F6 键使键盘的输入焦点在这两个窗口中移动。

在信息窗口中有一排高亮显示的句子,这句话是系统对程序中错误的一些提示,我们可以根据这些提示来查找错误。但是这些错误提示是以英语的形式出现的,对英语比较差的同学

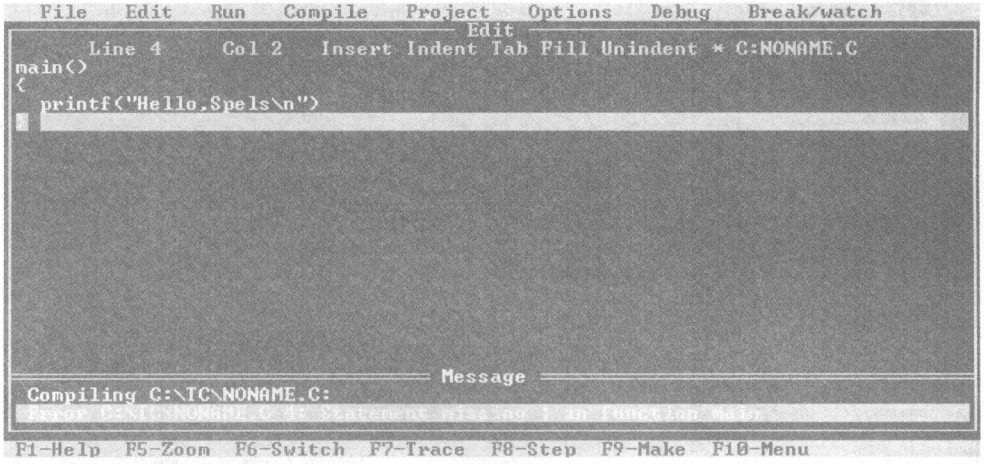

图 1-7　查错

来说，这简直是天书。所幸的是，我们不需要完全弄明白这句话的意思，下面介绍一些比较简单的方法去理解这些提示。

这句话的全部内容是：Error C:\TC\NONAME.C 4: Statement missing ; in function main。其中 Error 表示这是一个错误提示，如果是 Warning，则表示是一个警告提示，我们不用管它。紧接着的是一个含路径的文件名，也不用管它。再后面的数字 4，表示错误出现在第 4 行，也就是 Edit 窗口中高亮显示的那一行，因为在上面已经通过高亮显示提示了，所以也可以不用管。然后后面的一句话表示出错的提示了。这句话我们只要记住“missing ;”就行了，表示程序掉了分号。整个错误提示表示在程序的第 4 行掉了分号。

TC 的错误提示有以下两个重要的特点：

(1)提示中指出的错误出现的位置，不一定准确，但可以参考；

(2)实际的错误个数往往少于提示的错误个数。

在上例中，系统提示在程序的第 4 行掉了分号，但实际上第 4 行并没有掉分号，而是在第 3 行的末尾掉了分号。在程序的第 3 行末尾加上分号，程序就完全正确了。

有时候，系统提示有许多错误。这个时候不要被这么多错误吓倒了，实际的错误往往没有这么多。从第一个错误提示开始找出程序的错误，如果这个错误被找倒，不要再急于去找第二个错误。这时可以先把这个错误改正回来，然后按 Ctrl+F9 组合键。如何还有错误提示，则再去找新提示中的第一个错误提示。如此循环，直到没有错误为止。

查错是一个慢慢学习、慢慢积累经验的过程，不是一下子能全部记住的。本实验的附录一为部分常见的错误提示。这些提示并不需要马上记住，在编写程序出现错误的时候，可以对照附录一检查错误。多检查几次后，这些提示自然就被记住了。

7.保存

程序编写完成后，可以将程序的源代码保存到硬盘上。使用 File 菜单下的 Save 子菜单，或直接按 F2 键，出现保存对话框，如图 1-8 所示。

用退格键将框中的“C:\TC\NONAME.C”删除，输入文件名。文件名可以由你任意取名，但不得超过 8 个字符，也不得有空格。比如可以取名为 spels1，如图 1-9 所示。输入文件名后直接回车，文件就被保存了。这时打开 TC 文件夹，可以看到一个你取名的文件。

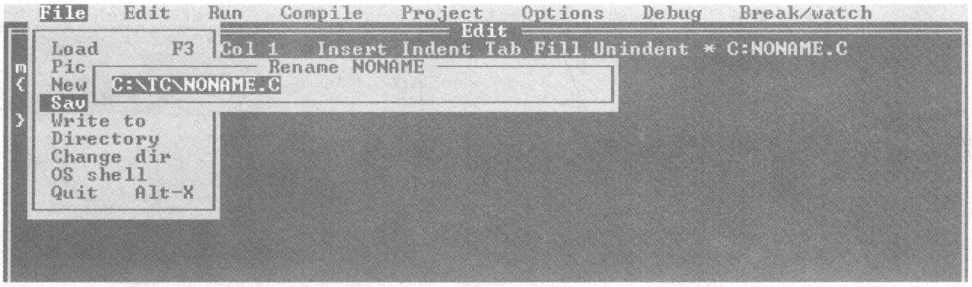

图 1-8　保存源程序

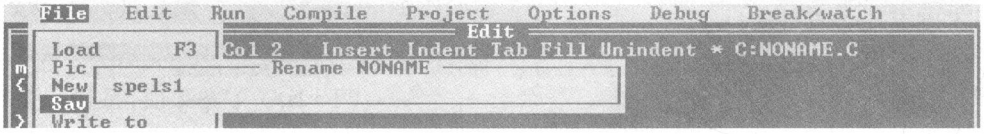

图 1-9　为文件取名

如果文件已经被保存过了，再使用 File 菜单下的 Save 菜单或直接按 F2 键，文件将被直接保存到原来所取名的文件中，这时在屏幕上没有任何反应。多次保存后，文件中存放的内容为最后一次保存的内容。

文件被保存后，在 Edit 窗口的右上角，会出现文件的名称，如图 1-10 所示。

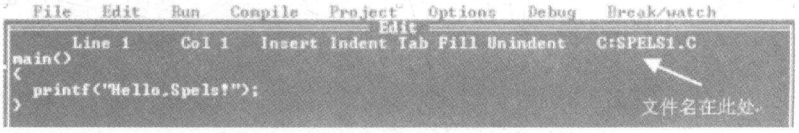

图 1-10　右上角出现文件名

如果保存文件时，与已经存在的文件名重名，则会出现提示信息“Overwrite SPELS1.C(Y/N)”，如果要用本文件覆盖原来的文件，按“Y”键；如果不想覆盖，按“N”键，再按 F2 键，重新取名，重新保存。

8.打开源程序

编写的源程序在被保存后，可以在以后需要的时候随时重新打开。打开文件可以使用 File 菜单下的 Load 子菜单，或直接按快捷键 F3。此时将出现一个打开文件对话框，如图 1-11 所示。

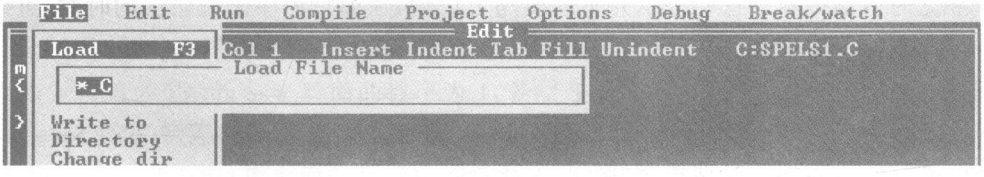

图 1-11　打开源程序

我们可以在这个对话框中输入想要打开的文件名，然后按回车键。其实，还有一个最简单

的办法，在出现上面的对话框后直接回车，出现如图 1－12 所示的界面。

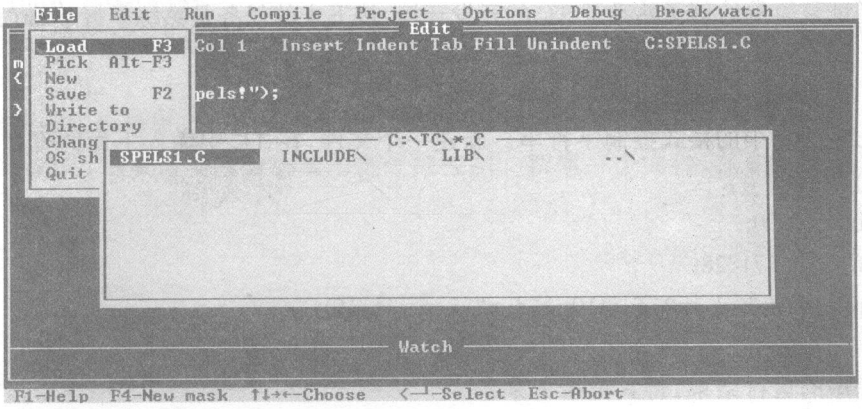

图 1－12　文件列表

注意：如果在文件对话框中出现的不是“* .C”，则直接将其内容全部删除，然后回车，也能得到如图 1－12 的界面。

在出现图 1－12 的界面后，使用方向键选择需要打开的文件，然后回车，就可以将文件打开了。

9.新建文件

当一个程序编写完成，需要重新编写第二个程序的时候，不能直接在第一个程序的后面编写，必须重新建立一个文件，然后编写。新建文件使用 File 菜单的 New 子菜单。

直接在第一个程序的后面编写第二个程序，这样在同一个源程序中会出现两个 main 函数，因此程序在编译时会出现错误。

10.退出 TC 系统

可以使用 File 菜单下的 Quit 子菜单退出 TC 系统。在退出 TC 系统时，如果文件没有被保存，则系统会提示“SPELS1.C not saved. Save? (Y/N)”，按“Y”将保存，按“N”键将不保存。

11.总结

编程的一般步骤：

① 编写程序；

② 按 Ctrl＋F9 运行程序；

③ 如果有错误，则检查程序中的错误，再执行第二步和第三步，直到没有错误为止；

④ 按 Alt＋F5 查看程序运行结果；

⑤ 按 F2 保存文件；

⑥ 使用 File 菜单下的 New 子菜单，新建一个文件，开始编写下一个程序。

常用快捷键见表 1－1。

表 1－1　常用快捷键

F10	激活主菜单	Ctrl＋ F9	编译＋ 链接＋ 运行
F9	编译＋ 链接	Alt＋ F5	看运行结果
F2	保存当前编写的程序	F3	打开源文件

二、理解顺序结构

前面通过最简单的例子熟悉了 TC 的开发环境，接下来的任务是熟练掌握 printf 函数和 scanf 函数。

1.练习 printf 函数

(1)练习 printf 中的格式控制字符串。新建一个文件，在 TC 中输入下面的程序代码。

```
main( )
{    int i=2000;
     float j=2.71828;
     printf("i=%d, j=%f, j*10=%f\n", i , j , j*  10);
}
```

查看程序的运行结果，将其记录下来。

① _____

我们将程序中 printf 函数那一行改为：

```
printf("j*10=%d, i=%f, j=%f\n",  i , j , j*10);
```

运行程序，记录下运行结果。

② _____

再将其改为：

```
printf("i  is %d, j  is %f, j*10   is %f\n", i, j , j*10);
```

运行程序，记录下运行结果。

③ _____

最后将其改为：

```
printf("%d, %f, %f\n", i, j, j*10);
```

运行程序，记录下运行结果。

④ _____

看看这 4 个结果有什么相同点和不同点，想想在 printf 函数的格式控制字符串(即双引号引起来的部分)中的 i 和 j 有什么作用。

结论：在格式控制字符串中出现的变量名，仅仅起一个提示的作用，它将被原样输出到屏幕上。其有无以及正确与否，均不影响后面值的输出。

(2)练习整数的不同格式控制符。新建一个文件，在 TC 中输入下面的程序代码。

```
main( )
{    int a=32767;
     printf("%d\n",a);
     printf("%o\n",a);
     printf("%x\n",a);
}
```

运行程序，看看相同的变量，在不同格式控制符的控制下，向屏幕输出不同的结果。

(3)练习 printf 函数对整型数据输出宽度的控制。新建一个文件，在 TC 中输入下面的程序代码。

```
main( )
{    int a=5732;
```

```
    printf("%d\n",a);
    printf("%6d\n",a);
    printf("%3d\n",a);
}
```

看看程序的运行结果，记住printf函数对整型数据输出宽度的控制规律。

(4)练习printf函数对实型数据输出宽度的控制。新建一个文件，在TC中输入下面的程序代码。

```
main( )
{   float  a=3.14159;
    printf("p=%10f\n",a);
    printf("p=%4f\n",a);
    printf("p= %.2f\n",a);
    printf("p=%.4f\n",a);
    printf("p=%2.4f\n",a);
    printf("p=%10.4f\n",a);
}
```

根据本实验记住printf函数对实型数据输出宽度的控制规律，注意对小数位数的舍弃情况。

(5)验证整型数据的存储结构。新建一个文件，在TC中输入下面的程序代码。

```
main( )
{   int a=-1;
    printf("%d\n",a);
    printf("%u\n",a);
}
```

用笔把-1在两个字节中的补码计算出来，再与程序第二个输出结果进行比较，想想为什么。

2.练习scanf函数

(1)练习使用scanf函数从键盘输入整型数据。新建一个文件，在TC中输入下面的程序代码。

```
main( )
{   int a;
    scanf("%d",&a);
    printf("%d\n",a*10);
}
```

运行程序，此时程序将停止在scanf函数中，等待用户的数据输入。因此，用户看到的是一个黑色的屏幕，光标在上面一闪一闪。如果不输入数据，则无论敲多少个回车键，程序均不能结束，也就是不能回到TC的界面。

在屏幕上输入45，再回车。程序运行结束，按Alt+F5查看运行结果。这时会发现在屏幕上输出了数字450，这就是我们要求程序输入的东西。多运行几次程序，每次输入不同的数字，看看运行结果。想想用scanf函数给变量赋值和直接给变量赋值有什么不同。

(2)练习使用 scanf 函数获取长整型数据。将上一个例子改为：

```
main( )
{    long a;
     scanf("% ld",&a);
     printf("% d\ n",a*10);
}
```

注意:% ld 中的 l 是字母,不是数字。运行程序,输入数据,看看程序的运行结果,然后将 scanf 函数中的格式控制符% ld 改为% d,即将 scanf 函数改为：

```
scanf("% d",&a);
```

运行程序,看看结果,想想为什么。

结论:使用 scanf 函数输入长整型数据时,必须使用% ld,不得省略字母 l。

(3)练习使用 scanf 函数获取单精度实型数据。新建一个文件,在 TC 中输入下面的程序代码。

```
main( )
{    float a;
     scanf("% f",&a);
     printf("% f\ n",a*10);
}
```

运行程序,输入数据,查看程序运行结果。

(4)练习使用 scanf 函数获取双精度实型数据。将上一个例子改为：

```
main( )
{    double a;
     scanf("% lf",&a);
     printf("% f\ n",a*10);
}
```

运行程序,输入数据。然后去掉% lf 中的字母 l,再运行程序,输入相同数据,查看程序运行结果。比较两次结果,想想为什么。

结论:使用 scanf 函数输入双精度实型数据时,必须使用% lf,不得省略字母 l。

(5)练习 printf 和 scanf 配合使用。新建一个文件,在 TC 中输入下面的程序代码。

```
main( )
{    int a;
     printf("Input a number:\ n");
     scanf("% d",&a);
     printf("% d\ n",a);
}
```

运行程序,输入数据,看看程序的运行结果。将程序第 3 行“printf("Input a number:\ n");”删除,再运行程序,输入相同数据,再看看程序的运行结果,想想为什么。

结论:scanf 函数前面的 printf 函数仅仅起到提示作用,其有无或具体是什么内容,均不影响后面的 scanf 函数。

(6)同时输入多个数据。新建一个文件,在 TC 中输入下面的程序代码。

```
main( )
{     int a;
      float b,c;
      scanf("% d% f% f",&a,&b,&c);
      printf("a=% d,b=% f,c=% f\ n",a,b,c);
}
```

注意:在 scanf 函数中%d 和两个%f 之间均无其他符号。运行程序,输入 3 个数据。注意这 3 个数据之间只能以空格、制表符号或回车符隔开,不能以逗号或其他符号隔开。

实验二 选择结构 循环结构

实验要求

1.熟悉选择结构的语法；
2.熟悉循环结构的语法，能编写简单的程序；
3.学习单步调试程序。

实验内容

一、选择结构

(1)新建一个文件，输入以下程序代码。

```
main( )
{     int a;
      scanf("% d",&a);
      if(a>=60)
            printf("Pass!\ n");
      else
            printf("Fail!\ n");
}
```

将程序多运行几遍，每次输入 0～100 之间不同的数据，看看程序的运行结果。

(2)新建一个文件，输入以下程序代码。

```
main( )
{     int a;
      scanf("% d",&a);
      if(a>=90)
            printf("A\ n");
      else if(a>=80)
            printf("B\ n");
      else if(a>=70)
            printf("C\ n");
      else if(a>=60)
```

```
            printf("D\n");
        else
            printf("E\n");
}
```

将程序多运行几遍，每次输入 0～100 之间不同的数据，看看程序的运行结果。

(3)新建一个文件，输入以下程序代码。

```
main( )
{    int    a=4,b=3,c=5,t;
     if(a<b) {t=a;a=b;b=t;}
     if(a<c) {t=a;a=c;c=t;}
     printf("%d %d %d\n",a,b,c);
}
```

运行程序，看看程序的运行结果。再将上面的程序改为：

```
main( )
{    int    a=4,b=3,c=5,t;
     if(a<b)   t=a;a=b;b=t;
     if(a<c)   t=a;a=c;c=t;
     printf("%d %d %d\n",a,b,c);
}
```

注意两个程序的区别，运行程序，看看程序的运行结果，比较一下两次结果的不同之处，分析一下原因。

二、循环结构

在编写循环结构程序的时候，往往由于忘记了对循环变量的值进行改变，从而变成了死循环。其症状是屏幕一直停留在 DOS 界面，不能做任何响应，这时候就需要强制将程序结束。按组合键 Ctrl+Break 可以强制结束程序。

(1)求出 1～100 之间的和。新建一个文件，输入以下程序代码。

```
main( )
{    int i=0,s=0;
     while(i<=100)
     {   s+=i;
         i++ ;
     }
     printf("%d\n",s);
}
```

运行程序，查看程序的运行结果。将循环判断条件改为 i<100，再运行程序，看看程序的运行结果，分析一下原因。

试着将上面的程序改为 do-while 循环结构和 for 循环结构，分别运行程序，查看程序运行结果。

(2)新建一个文件，输入以下程序代码。

```
main( )
{    int i=1,sum=1;
     while(i<=5)
     {  sum=sum*i;
        i++ ;
     }
     printf("% d",sum);
}
```

运行程序，查看程序的运行结果，分析程序实现的功能。再将上面的程序改为 do-while 循环和 for 循环。

(3)新建一个文件，输入以下程序代码。

```
main( )
{    int a;
     do
     {  scanf("% d",&a);
        if(a>=60)
           printf("Pass!\ n");
        else
           printf("Fail!\ n");
     }while(a!=0)
}
```

运行程序，输入数据，看看程序的运行结果。当输入数据 0 时，结束程序的运行。

(4)利用公式 π/4=1－1/3＋1/5－1/7＋1/9－…求 π 的近似值，直到最后一项的绝对值小于 10^{-4} 为止。新建一个文件，输入以下程序代码。

```
# include "math.h"
main( )
{    int i=1,a=1;
     float pi= 0,s= a*1.0/i;
     while(fabs(s)>=1e－4)
     {  pi+=s;
        a=－a;
        i+=2;
        s=a*1.0/i;
     }
     pi*=4;
     printf("% f\ n",pi);
}
```

说明：fabs 是求绝对值的函数，调用 fabs 函数必须包含头文件 math.h。运行上面的程序，查看运行结果，思考并验证以下几个问题。

①变量 a 起什么作用？