



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

程序设计基础实验 及学习指导

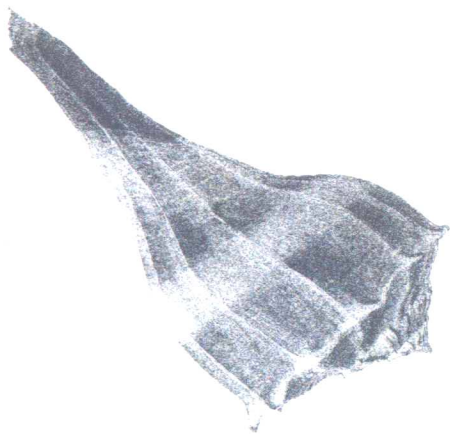
Guide of Experiment and Study for
Fundamentals of Programming

高伟 主编

丛晓红 副主编

李江华 主审

- 自主研发实验，引领学生思维
- 注重边练边学，变被动为主动
- 强调实践中学，变失败为成功



高校系列



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化部普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

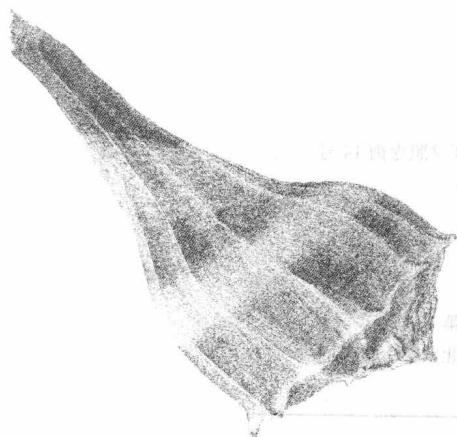
程序设计基础实验 及学习指导

Guide of Experiment and Study for Fundamentals of Programming

高伟 主编

丛晓红 副主编

李江华 主审



高校系列

人民邮电出版社

北京

图书在版编目 (CIP) 数据

程序设计基础实验及学习指导 / 高伟主编. — 北京:
人民邮电出版社, 2012.3
21世纪高等学校计算机规划教材
ISBN 978-7-115-27373-4

I. ①程… II. ①高… III. ①程序设计—高等学校—
教学参考资料 IV. ①TP311.1

中国版本图书馆CIP数据核字(2011)第282216号

内 容 提 要

本书为程序设计基础课程的配套辅助教材,用于实验和自学环节,可以在学生学习C语言编写程序的同时,做到即学即练,即练即用,从而达到举一反三、触类旁通、深刻理解程序设计理念的目的。

本书包括4章:实验指导、同步习题与习题解析、常用编程算法和模拟试题。全书采用Turbo C 3.0作为编译器。

本书注重让学生从练中学,通过安排“基础”、“设计”和“自主研发”3个层次的实验,逐步提升学生的能力,用“指导”的形式思维指点方向。

本书适合作为高等学校非计算机专业学生学习程序设计基础课程的辅助教材,也可作为全国计算机等级考试的参考书。

工业和信息化部普通高等教育“十二五”规划教材立项项目

21世纪高等学校计算机规划教材

程序设计基础实验及学习指导

◆ 主 编 高 伟

副 主 编 丛晓红

主 审 李江华

责任编辑 武恩玉

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京天宇星印刷厂印刷

◆ 开本: 787×1092 1/16

印张: 13.5

2012年3月第1版

字数: 356千字

2012年3月北京第1次印刷

ISBN 978-7-115-27373-4

定价: 27.00元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前言

程序设计基础是教育部高等学校计算机基础课程教学指导委员会制定的《高等学校计算机基础核心课程教学实施方案》中的核心课程,其教学目标定位于“通过介绍一种具体的程序设计语言及其程序设计方法,使学生了解程序设计语言的基本结构,理解计算机学科求解实际问题的基本过程,掌握程序设计的基本过程,掌握程序设计的基本思想、方法和技巧,养成良好的程序设计风格,培养利用计算机求解问题的基本能力”。我们本着上述指导思想,编写了本书。

本书包含4章。第1章为实验指导,讲解编程环境和实验方法,通过9个实验,使学生理解所学知识及编程方法。第2章为配套习题,通过习题解析使学生分章节理解C语言及编程技术。第3章为常用编程算法,目的在于提高编程能力和掌握常用技巧,扩展视野。第4章为模拟试题,可使学生自行检验自己的学习效果,训练综合能力。

实验部分采用Turbo C 3.0作为编译器,因其采用ANSI C 99标准,且小巧灵活,操作简单易学,既能完成编程任务,又支持鼠标操作。

教材编写强调理论和实践相结合,学习和能力相结合,应用和创新相结合。在实验部分通过基础实验、设计实验和自主研发实验3个层次由浅入深地引领学生的学习思路。基础实验用来领会所学的理论知识;设计实验侧重理论知识的灵活运用;自主研发实验用于激发创新思维。为打破初学者思路的局限性,用“指导”的形式思维指点方向。这种写作模式符合在计算机基础课程中培养学生“计算思维”的先进教学理念。

本书由具有多年基础教学经验的一线教师编写,体现了教学团队的整体力量,力求把教学中的心得体会融入到教材之中,着重于让学生易于理解,提高学生的自主学习能力。本书由高伟任主编,丛晓红任副主编,李江华任主审,崔玉文、董宇欣、郭江鸿、宁慧、苏哲明、唐立群、魏传宝、吴良杰、徐丽、赵宝刚(按字母音序排列)共同编写。

本书在写作过程中得到了哈尔滨工程大学教学督导孙长嵩教授的鼎力支持,衷心感谢孙教授对本书提出的宝贵意见。在此也一并感谢其他支持本书写作的同事及所有对本书做出贡献的人。

编者

2011年11月

第 1 章 程序设计基础

实验指导 1

1.1 熟悉 C 程序的运行环境及操作实验.....	1
1.2 顺序结构程序设计实验.....	9
1.3 选择结构程序设计实验.....	16
1.4 循环程序设计实验.....	25
1.5 函数实验.....	31
1.6 数组实验.....	38
1.7 结构体实验.....	45
1.8 指针实验.....	50
1.9 文件操作实验.....	56

第 2 章 程序设计同步练习指导 62

2.1 程序设计基础.....	62
2.1.1 要点指导.....	62
2.1.2 同步练习.....	62
2.2 数据表示及数据运算.....	63
2.2.1 要点指导.....	63
2.2.2 同步练习.....	65
2.3 最简单的 C 程序设计.....	67
2.3.1 要点指导.....	67
2.3.2 同步练习.....	68
2.4 选择结构程序设计.....	69
2.4.1 要点指导.....	69
2.4.2 同步练习.....	70
2.5 循环结构程序设计.....	75
2.5.1 要点指导.....	75
2.5.2 同步练习.....	76
2.6 函数.....	84
2.6.1 要点指导.....	84
2.6.2 同步练习.....	86
2.7 数组.....	93

2.7.1 要点指导.....	93
-----------------	----

2.7.2 同步练习.....	95
-----------------	----

2.8 常见算法.....	105
---------------	-----

2.9 结构体.....	105
--------------	-----

2.9.1 要点指导.....	105
-----------------	-----

2.9.2 同步练习.....	105
-----------------	-----

2.10 指针.....	107
--------------	-----

2.10.1 要点指导.....	107
------------------	-----

2.10.2 同步练习.....	108
------------------	-----

2.11 文件.....	134
--------------	-----

2.11.1 要点指导.....	134
------------------	-----

2.11.2 同步练习.....	134
------------------	-----

第 3 章 常用算法 136

3.1 程序设计基础.....	136
3.2 数据表示及数据运算.....	136
3.3 最简单的 C 程序设计.....	137
3.4 选择结构程序设计.....	138
3.5 循环结构程序设计.....	142
3.6 函数.....	147
3.7 数组.....	151
3.8 常见算法.....	164
3.9 结构体.....	165
3.10 指针.....	172
3.11 文件.....	178

第 4 章 考试样题 181

4.1 考试样题一.....	181
4.2 考试样题二.....	187
4.3 考试样题三.....	193
4.4 考试样题四.....	199
4.5 考试样题五.....	205

第 1 章

程序设计基础实验指导

说明：本章的每一节都包含 4 个部分——基础实验、设计实验、自主研发实验和实测演练。基础实验用以帮助对所学内容的理解；设计实验是在基础实验的基础上设计指定题目的程序；自主研发实验是在一定基础上自行拟定题目自主编程，也可以根据指导的内容编程；实测演练用来测试基础知识的掌握情况和分析问题能力、推理能力及综合能力。

1.1 熟悉 C 程序的运行环境及操作实验

一、实验目的

1. 了解在 PC 机上如何编辑、编译、链接和运行程序。
2. 学习 Turbo C++ 3.0（简称 TC）集成开发环境的使用。

二、基础实验

实验 1：使用一个小程序来练习在 TC 环境下编辑、编译、链接和运行程序。练习只有输出的程序。程序的功能是在屏幕上输出一行文字“This is a C program.”并回车。

实验步骤：

第 1 步：打开 TC 编辑器，并将编辑器默认的源程序扩展名设置为.C。

双击桌面上的 TC 快捷方式图标，如图 1.1 所示。

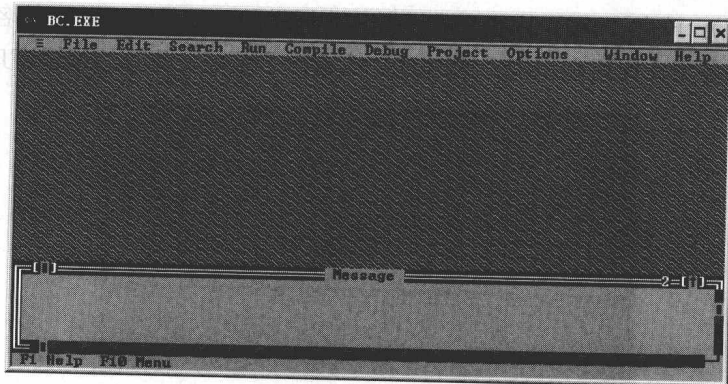


图 1.1 TC 程序窗口

执行“Options”菜单的“Environment”子菜单的“Editor...”命令，打开“Editor Options”（编辑器选项）设置对话框。在“Default Extension”（默认扩展名）文本框中输入“C”，单击“OK”按钮完成设置，如图 1.2 所示。

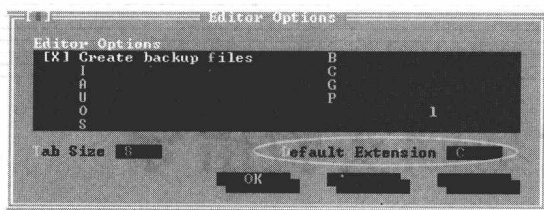


图 1.2 配置源程序默认扩展名

第 2 步：编辑程序。

执行“File”菜单的“New”命令，新建一个名为“NONAME00.C”的源文件（注意：文件名在编辑窗口的中间），如图 1.3 所示。

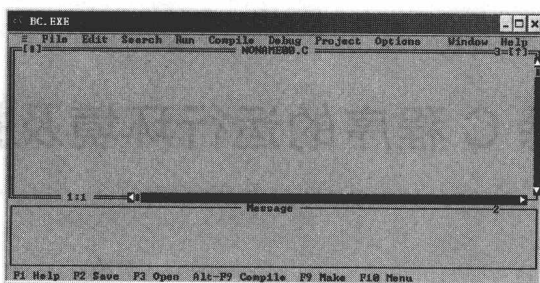


图 1.3 新建源文件窗口

在编辑窗口中输入下述程序：

```
#include <stdio.h>          /* 第 1 行 */
void main( )                /* 第 2 行 */
{                             /* 第 3 行 */
    printf("This is a C program.\n"); /* 第 4 行 */
}                             /* 第 5 行 */
```

执行“File”菜单的“Save as...”命令，打开“Save File As”（另存为）对话框，在“Save File As”文本框中输入“L1”（因为在前面已将编辑器的默认源文件扩展名设置为“.C”，所以此处可以只输入主文件名），单击“OK”按钮完成设置。这样做的目的是将当前文件名由默认名改为自定义的文件名，以备用户保留和识别，同时对文件进行了保存。此时的编辑窗口如图 1.4 所示。

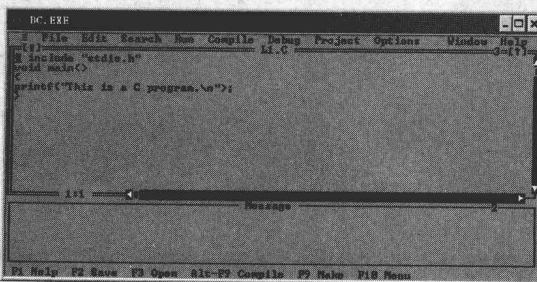


图 1.4 编辑窗口

第3步：编译、链接程序。

执行“Compile”菜单的“Compile”命令进行编译。为了提高操作速度，执行此步操作时可用组合键“Alt+F9”。若无编译错误，则弹出编译成功对话框，如图1.5所示。看到编译成功提示后，按键盘上的任意一个按键，操作返回到编辑界面。

执行“Compile”菜单的“Link”命令进行链接。若无链接错误，则弹出链接成功对话框，如图1.6所示。看到链接成功提示后，按键盘上的任意一个按键，操作返回到编辑界面。

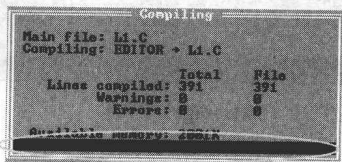


图 1.5 编译成功对话框

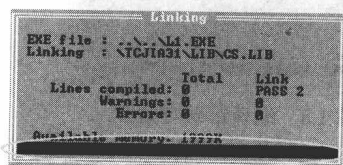


图 1.6 链接成功对话框

第4步：运行程序。

执行“Run”菜单的“Run”命令来运行程序。也可用组合键“Ctrl+F9”来完成此操作。此步操作完成后在屏幕上没有任何提示。

用“Windows”菜单的“User screen”命令来查看执行结果。也可用组合键“Alt+F5”来完成此操作。这时控制被切换到运行结果窗口，可见屏幕上出现了预期的一行文字。如图1.7所示。



图 1.7 运行结果窗口

可按任意键返回到编辑窗口。至此该程序已调试成功。在编辑窗口执行“File”菜单的“Quit”命令可关闭 TC 程序。可用组合键“Alt+X”来完成此操作。

注意事项：

- (1) 每个程序的第一行必须是：`# include <stdio.h>`。
- (2) `main()`函数前必须加 `void`。
- (3) 所有引号及括号必须成对出现。
- (4) 语句的后面必须用分号。
- (5) 为主文件命名时最好做到“见名知意”。主文件名长度不能超过8个字符，且不能包含小数点和空格及非法字符：`\`，`/`，`<`，`>`，`|`。
- (6) 编译之前一定要先保存文件，以防止误操作而丢失信息。
- (7) 程序可被编译、链接和运行多次。一定要学会使用快捷键以提高操作速度。
- (8) 当程序编辑完成后没进行编译之前执行运行命令，系统会自动按照编译、链接、运行的顺序执行。因此当达到熟练编辑，且错误很少的程度时，建议直接使用运行命令“Ctrl+F9”来提高操作速度。

实验2：在前例的基础上进行简单调试练习，进一步熟悉软件的使用。

前一个实验是在程序无编辑错误的前提下完成的，但实际情况并非如此。下面人为设置一些错误，请认真记录错误提示，以提高程序调试能力。参考图1.4，删掉第4行的最后一个符号——分号，按“Ctrl+F9”，运行程序，弹出编译错误对话框，如图1.8所示。

按任意键关闭该窗口，在编辑窗口下方弹出“Message”窗口（信息窗口），如图 1.9 所示。

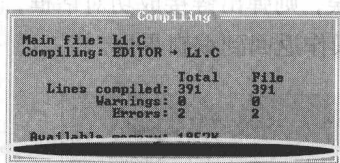


图 1.8 编译错误对话框

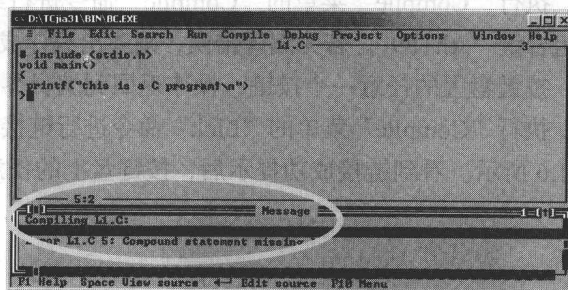


图 1.9 “Message”窗口

3 条编译信息的含义如下。

“Compiling L1.C:”——正在编译源文件 L1.C。

“Error L1.C 5: Statement missing ;”——在源文件 L1.C 的第 5 行发现错误：语句缺分号。

“Error L1.C 5: Compound statement missing)”——复合语句缺圆括号。

在调试程序的过程中经常有各种错误。在设计错误时，被删掉的是第 4 行语句后的分号，但系统给出的提示是：在第 5 行发现语句缺分号错误。出现这种现象是因为 C 支持程序自由书写风格，允许将一个语句写在多行上。当系统在第 4 行没有找到分号时，认为分号应该出现在第 5 行并继续向下扫描第 5 行。在第 5 行仍然没有找到分号时，才给出错误提示：第 5 行缺分号。

用鼠标点一下编辑窗口，或按“F6”键切换到编辑窗口，补上被删掉的分号。重新按“Ctrl+F9”运行程序。此时没有弹出其他窗口，但“Message”窗口中的内容发生了变化，表明该程序已通过编译，链接并被执行了。可随时按“Alt+F5”查看运行结果。如图 1.10 所示。

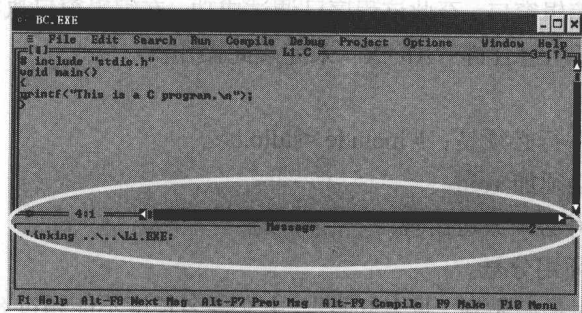


图 1.10 无编译及链接错误的状态

仿照上述操作方法，参考图 1.4，自行作几个验证性实验：

(1) 将第 2 行的“main”改成“mian”。

(2) 删掉第 4 行“printf”中的“f”。

(3) 删掉第 5 行的花括号。

(4) 删掉第 1 行的“#”号。

(5) 删掉第 4 行中的圆括号“（”。

(6) 删掉第 4 行中的引号。

实验 3：练习有计算并输出的程序。设变量 a 的值为 3，变量 b 的值为 5，求和并送入变量 c，显示 c 的值。将 a 的值改为 100，b 的值改为 200，重新执行程序，观察运行结果。

新建一个源文件窗口，输入下面的程序：

```
# include <stdio.h>
void main( )
{ int a,b,c;          /* 定义3个整型变量 a, b, c */
  a=3;                /* 给 a, b 赋值 */
  b=5;
  c=a+b;              /* 求和 */
  printf("c=%d\n",c); /* 输出结果 */
}
```

执行“File”菜单的“Save as ...”命令，将文件另存为“L2.C”。按“Ctrl+F9”运行程序。按“Alt+F5”，在运行结果窗口可看到“c=8”，表示本次运行成功。

修改程序的第4行，把“3”改成“100”。把第5行中的“5”改成“200”。再次按“Ctrl+F9”运行程序。按“Alt+F5”，在运行结果窗口可看到“c=300”，表示本次运行成功。

实验4：练习有输入、计算并输出的程序。设变量a的值为3，变量b的值由键盘输入，求和并送入变量c，显示c的值。

新建一个源文件窗口，输入下面的程序：

```
# include <stdio.h>
void main( )
{ int a,b,c;          /* 定义3个整型变量 a, b, c */
  a=3;                /* 给 a 赋值 */
  scanf("%d",&b);      /* 通过键盘给 b 赋值 */
  c=a+b;              /* 求和 */
  printf("c=%d\n",c); /* 输出结果 */
}
```

执行“File”菜单的“Save as ...”命令，将文件另存为“L3.C”。按“Ctrl+F9”运行程序。这时弹出运行窗口，并且屏幕上没有任何数据，这种状态表示等待输入b的值。输入“5”并回车。系统关闭运行窗口，返回到了编辑状态。按“Alt+F5”查看运行结果。第1行是输入的值，第2行是输出的值，如图1.11所示。

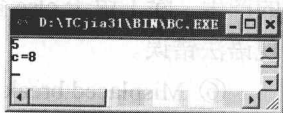


图 1.11 运行结果

如果想计算3和另外一个数的和，就可以再次执行程序，在运行时输入数据，而不必修改程序。

重要提示：

(1) 用“File”菜单的“Quit”命令退出TC。一般情况下，切勿通过单击“关闭”按钮的方法退出TC，因为这种操作可能导致数据的丢失。

(2) 注意光标表示的输入状态。光标为闪烁的短线时为“插入”状态，光标为闪烁的矩形时为“改写”状态。

(3) 在操作时，可以利用快捷键加快操作速度。若某菜单命令有快捷键，则会显示在其右侧。常用的几个组合快捷键如下。

- ① Ctrl+F9：运行。
- ② Alt+F5：查看结果。
- ③ Alt+X：退出TC。
- ④ Insert：切换插入/改写状态。

- ⑤ F1: 获取帮助。
- ⑥ F6: 在各窗口之间进行切换。
- ⑦ Alt+F3: 关闭窗口。
- ⑧ Alt+数字: 激活数字代表的窗口。
- ⑨ F8: 单步执行程序。
- ⑩ Ctrl+F7: 设置观察量。
- ⑪ ESC: 返回上一菜单。
- ⑫ Ctrl+Break: 运行时, 中断死循环状态, 返回到编辑状态。

(4) 在编译时会有 3 类常见的错误: 致命错误、一般错误、警告。致命错误很少, 通常是内部编译出错。在发生错误时, 立即停止, 必须采取一些适当的措施并重新编译。一般错误指程序的语法错误、磁盘或内存存取错误或命令错误等。编译程序在每个阶段(预处理、语法分析、优化和代码生成)都尽可能多地发现源程序中的错误。警告并不阻止编译进行。它指出一些值得怀疑的情况, 而这些情况本身又有可能合理地成为源程序的一部分。如果在源文件中使用了与机器有关的结构, 编译也将产生警告信息。

有如下几个常见的编译信息。

- ① Statement missing ;: 语句缺少 “;”。编译程序发现一些表达语句后面没有分号。
- ② undefined symbol xxxxxxxx: 符号 xxxxxxxx 未定义。标识符无定义, 可能是由于说明或引用处有拼写错误, 也可能是由于标识说明错误引起。
- ③ unterminated string or character constant: 未终结的串或字符常量。编译程序发现串或字符常量开始后没有终结。
- ④ User break: 用户中断。在集成环境里进行编译或链接时用户按了 “Ctrl+Break” 键。
- ⑤ Misplaced else: else 位置错误。编译程序发现 else 语句缺少与之匹配的 if 语句。此类错误的产生, 除了由于 else 多余外, 还有可能是由于有多余的分号, 漏写了大括号或前面的 if 语句出现语法错误。
- ⑥ Misplaced break: break 位置错误。编辑程序发现 break 语句在 switch 语句或循环结构外。
- ⑦ Expression syntax: 表达式语法错误。当编译程序分析一表达式发现一些严重错误时, 出现此类错误, 通常是由于两个连续操作符、括号不匹配或缺少括号, 前一语句漏掉了分号等引起的。
- ⑧ function call missing): 函数调用缺少 “)”。
- ⑨ default outside of switch: default 在 switch 外出现。编译程序发现 default 语句出现在 switch 语句之外, 通常是由于括号不匹配造成的。
- ⑩ division by zero: 除数为 0。源文件的常量表达式中, 出现除数为 0 的情况。
- ⑪ do statement must have while : do 语句中必须有 while。源文件中包含一个无 while 关键字的 do 语句时, 出现此类错误。
- ⑫ compound statement missing: 复合语句漏掉了大括号 “}”。编译程序扫描到文件时, 未发现结束大括号, 通常是由于大括号不匹配造成的。
- ⑬ Illegal use of floating point: 浮点运算非法。
- ⑭ Redeclaration of 'xxxxxxx': ‘xxxxxxx’ 重定义。此标志符已经定义过。
- ⑮ Subscripting missing]: 下标缺少 “]”。编译程序发现一个下标表达式右方括号, 可能是由于漏掉或多写操作符或括号不匹配引起的。

- ### 三、设计实验

四、自主研发实验

五、实测演练

- (2) C 语句的结束符是_____。
- (3) C 的程序有且仅有一个_____函数。
- (4) C 函数的 3 个组成部分是_____、_____和_____，其中_____是必不可少的。

(5) 将数学式 $\frac{1+\frac{1}{b}}{2c}$ 写成 C 的表达式。

2. 选择题

- (1) 下列标识符中，合法的用户标识符是_____。
- A. 1a2b3c B. _123 C. a?1 D. int
- (2) 如果下列变量都是整型的，且 sub=pad=5，pAd=sum++，pAd++，++pAd，则语句 printf("%d",pad); 的输出结果是_____。
- A. 5 B. 6 C. 7 D. 8
- (3) 以下表达式的值是 3 的是_____。
- A. 16-13%10 B. 2+3/2 C. 14/3-2 D. (2+6)/(12-9)

3. 改错题 (改正注释所在行的错误)

```
#include <stdio.>          /* Found */
void mian()                /* Found */
{ Int a,b;                 /* Found */
    scanf("%d",a);         /* Found */
    scanf("d%",&b);        /* Found */
    a+=b;                  /* Found */
    printf("a=%d,a);      /* Found */
}
```

参考答案

1. 填空题

- (1) 顺序结构，选择结构，循环结构 (2) ; (3) main()
- (4) 输入部分，计算部分 (执行部分)，输出部分；输出部分 (5) (1+1/a*b)/(2*c)

2. 选择题

- (1) B (2) A (3) B

3. 改错题

```
#include <stdio.h>          /* 头文件的扩展名为.h */
void main( )                /* 主函数名为 main( ) */
{ int a,b;                  /* 大小写是不同的字母 */
    scanf("%d",&a);         /* 输入变量值时，必须用变量地址 */
    scanf("%d",&b);        /* 格式控制符由%开头 */
    a+=b;                   /* 语句必须用；结束 */
    printf("a=%d",a);      /* "必须成对出现 */
}
```

六、设计实验参考程序

实验 5:

```
#include <stdio.h>
void main( )
```

```
{ printf("*****\n");
  printf("*   This is my first C program!   *\n");
  printf("*****\n");
}
```

实验 6:

```
#include <stdio.h>
void main( )
{ printf("*****\n");
  printf("* 100001, Bei Jing           *\n");
  printf("*                               *\n");
  printf("*      Tian An Men You Ju         *\n");
  printf("*                               *\n");
  printf("*                               HRBEU,150001 *\n");
  printf("*****\n");
}
```

实验 7:

```
#include <stdio.h>
void main( )
{ printf(" 88888888\n");
  printf(" 88      88\n");
  printf("8  @@  8\n");
  printf("8      8\n");
  printf(" 88  00  88\n");
  printf(" 88888888\n");
}
```

实验 8:

```
#include <stdio.h>
void main( )
{ int a,b,c;
  a=200;b=30;
  c=a*b;
  printf("c=%d\n",c);
}
```

实验 9:

```
#include <stdio.h>
void main( )
{ int a,b,c;
  a=100;
  scanf("%d",&b);
  c=b-a;
  printf("c=%d\n",c);
}
```

1.2 顺序结构程序设计实验

一、实验目的

1. 了解 C 语言数据类型, 熟悉如何定义整型、字符型、实型变量以及对它们进行赋值的方法。

2. 熟悉算术运算符、自加自减运算符和算术表达式的使用；熟悉赋值运算符、赋值表达式的使用。
3. 掌握调试程序的基本方法。
4. 掌握基本输入输出方法，正确使用格式符。
5. 熟悉并使用 scanf、printf 函数进行输入输出。
6. 了解文件包含的初步概念，学会使用标准函数。

二、基础实验

实验 1：给变量赋初值，进行整型与实型数据的混合运算，运行程序，并分析结果。

新建一个源文件窗口，输入下面的程序：

```
#include <stdio.h>
void main( )
{ int a,b;                                /* 定义各变量 */
  char x,y;
  float num,u;
  a=b=100;                                /* 给各变量赋值 */
  x=y='A';
  num=u=3.6792;
  a=y;                                    /* 进行混合运算 */
  x=b;
  num=b;
  a=a+u;
  printf("a=%d,x=%c,num=%f,a=%d",a,x,num,a); /* 输出执行结果 */
}
```

执行“File”菜单的“Save as...”命令，将文件另存为“L4.C”。按“Ctrl+F9”运行程序。这时弹出结果为“a=68, x=d, num=100.000000, a=68”。请先分析程序，推导出输出结果。

程序说明：第 3 行定义 2 个整型变量 a, b。第 4 行定义 2 个字符型变量 x, y。第 5 行定义 2 个实型变量 num, u。第 6 行为整型变量赋整型常量 100。第 7 行为字符型变量赋字符常量'A'。第 8 行为实型变量赋实型常量 3.6792。第 9 行用字符型变量 y 为整型变量 a 赋值，此时系统进行数据类型转换，将 1 个字节的字符'A'的 ASCII 值转换成 2 个字节的整数 65，然后赋值给变量 a。第 10 行用整型变量 b 为字符型变量 x 赋值，此时系统进行数据类型转换，将 2 个字节的整型常数 100 的低字节的值 100 作为 ASCII 值（对应字母'd'）赋值给变量 x。第 11 行用整型变量 b 为实型变量 num 赋值，此时系统进行数据类型转换，将 2 个字节的整型常数 100 转换成实型常量 100.0，然后赋值给变量 num。第 12 行用表达式 a+u 的值为整型变量 a 赋值，系统先计算 65+3.6792，结果为 68.6792，再转换成整型 68 赋给整型变量 a。

修改程序的倒数第 2 行为：printf("a=%c,x=%d,num=%d,a=%f",a,x,num,a);。执行“File”菜单的“Save as...”命令，将文件另存为“L4-1.C”。按“Ctrl+F9”运行程序。这时弹出结果为“a=D, x=100, num=0, a=0.000000”。请先分析程序，推导出输出结果，再将推导结果与实际运行结果作对比，对推导过程进行验证，并与前次运行结果做对比。

程序说明：修改程序后各变量的值没有变，改变的仅仅是数据的输出格式。由于字符型量在存储时，存储的是整数形式的 ASCII 值，所以 C 规定整型与字符型是通用的。因此将值为 68 的整型变量 a 以 C 格式输出时，输出的是 ASCII 值 68 对应的字符'D'；将值为'd'的字符型变量 x 以 d

格式输出时,输出的是字符'd'对应的 ASCII 值 100。C 没有规定整型数据与实型数据输出时的转换关系,所以将 4 个字节的实型量 num 作为 2 个字节的整型量输出时或者将 2 个字节的整型量 a 作为 4 个字节的实型量输出时,系统给出的数据都是不准确的。

由于上述原因,在写输出语句时一定要准确使用输出格式。

实验 2: 设 a 的值为 12, 编程计算 3 个复合赋值表达式 $a+=a$, $a/=a+a$, $a+=a-a*=a$ 的值。

参考程序如下:

```
#include <stdio.h>           //第1行
void main( )                 //第2行
{ int a=12,a1,a2,a3;         //第3行
  a1=(a+=a);                 //第4行
  a=12;                       //第5行
  a2=(a/=a+a);               //第6行
  a=12;                       //第7行
  a3=(a+=a-a*=a);           //第8行
  printf("%d %d %d ",a1,a2,a3); //第9行
}
```

运行程序,分析运行结果。请回答下面问题:

- (1) 执行完第 4 行后,变量 a 的值是多少?
- (2) 执行完第 6 行后,变量 a 的值是多少?
- (3) 执行完第 8 行后,变量 a 的值是多少?
- (4) 第 5 行和第 7 行的语句 ($a=12$;) 作用是什么?
- (5) 删除第 5 行和第 7 行的语句,运行结果又如何?

实验 3: 理解自加自减运算。

参考程序如下:

```
#include <stdio.h>
void main( )
{ int i,j,m,n;
  i=8;j=10;
  m=i++;n=j++;
  printf("%d,%d,%d,%d",i,j,m,n);
}
```

运行程序,分析运行结果。请回答下面问题:

- (1) 将第 5 行的“j++”改为“++j”对运行结果是否有影响? 新的运行结果是什么?
- (2) 将第 5 行改为“m=i;n=j; i++;j++;”后运行。分析运行结果与初次结果的异同。
- (3) 将第 5 行改为“i++;j++;m=i;n=j;”后运行。分析运行结果与初次结果的异同。
- (4) 将第 5 行改为“m=i;n=j; ++i; ++j;”后运行。分析运行结果与初次结果的异同。
- (5) 再提 2 种方案对第 5 行作修改,使其与初始程序等价。
- (6) 对自加自减运算进行归纳和总结。

实验 4: 理解“整型与字符型通用”这句话的含义。

参考程序如下:

```
#include <stdio.h>
void main( )
```

```
{ int c1,c2;
  char i1,i2;
  c1=65;
  c2=97;
  i1='A';
  i2='a';
  printf("%c %c\n",c1,c2);
  printf("%d %d\n",i1,i2);
}
```

请分析运行结果，并实际运行程序进行验证。

实验 5: 理解 printf 函数的使用。程序的功能是输入一个十进制数，输出该数和对应的十六进制数和八进制数。运行时输入 32767。

参考程序如下：

```
#include <stdio.h>
void main( )
{ int a;
  printf("Please Input a Integer: ");
  scanf("%d",&a);
  printf("shi jin zhi %d de shi liu jin zhi shi %x,ba jin zhi shi %o\n",a,a,a);
}
```

注意事项：

(1) 程序执行到第 4 行时，在显示器上输出字符串“Please Input a Integer:”，作用是提示用户下面的操作是要从键盘输入数据。可以用这种方法提示用户当前应该进行什么操作。应习惯于采用这种编程风格。

(2) 注意第 5 行语句中变量 a 之前的取地址运算符“&”一定不要漏掉，因为 scanf 函数要求传送变量的地址。

(3) 由于 TC 不支持汉字，所以第 6 行是用汉语拼音做的提示，请仔细阅读。

(4) 手工计算 32767 的十六进制数和八进制数，并与运行结果作对比。

实验 6: 练习使用数学函数。已知表示双精度数的变量 $\pi=3.14159265$ ，编写求 π 的平方根、 $-\pi$ 的绝对值和 π 的正弦函数值的程序。

提示：

(1) 使用数学函数时，要包含头文件 math.h。

(2) 平方根函数为 sqrt，绝对值函数为 fabs，正弦函数为 sin。

参考程序如下：

```
#include <stdio.h>
#include <math.h>
void main( )
{ double pi=3.1415926;
  printf("ping fang gen =%f\n",sqrt(pi));          /* 用%f 格式输出双精度数 */
  printf("-pi de jue dui zhi =%f\n",fabs(-pi));
  printf("zheng xian zhi =%f\n",sin(pi));
}
```

实验 7: 运行下述程序，分析带符号整数与无符号整数的关系。

```
#include <stdio.h>
void main( )
{ int a=100;
```