



快乐大本·优秀教材辅导
KUAILE DABEN
YOUXIUJIAOCAIFUDAO

大学计算机(C语言) 实验及学习指导

主 编 高 伟

- 课后习题 精析 精解
- 同步训练 勤学 勤练

SHIYAN
JI XUEXI ZHIDAO

HEUP 哈尔滨工程大学出版社
Harbin Engineering University Press



快乐大本 优秀教材辅导
KUAILE DABEN
YOUXIUJIAOCAIFUDAO

大学计算机(C语言) 实验及学习指导

主 编 高 伟

副主编 丛晓红

主 审 李江华

HEUP 哈尔滨工程大学出版社
Harbin Engineering University Press

内 容 简 介

本书是基于高等院校计算机基础课程 21 世纪教育改革方案的思想而编写的 C 语言辅导教材。本书主要内容包括如何使用 Turbo C 3.0 编辑、编译、链接、运行和调试 C 程序以及各章节的同步练习和解答。

本书大量习题均选自全国计算机等级考试二级 C 语言和省计算机等级考试二级 C 语言中的试题。

本书适合作为高等院校学生学习 C 语言的同步配套教材,特别适合作为学生学习 C 语言后参加全国计算机等级考试的辅助教材。本书也适合作为自学 C 语言的配套教材。

图书在版编目(CIP)数据

大学计算机(C语言)实验及学习指导/高伟主编. —哈尔滨:
哈尔滨工程大学出版社,2011.3
ISBN 978-7-5661-0078-8

I. ①大… II. ①高… III. ①C 语言-程序设计-高等学校-
教学参考资料 IV. ①TP312

中国版本图书馆 CIP 数据核字(2011)第 034275 号

出版发行 哈尔滨工程大学出版社
社 址 哈尔滨市南岗区东大直街 124 号
邮政编码 150001
发行电话 0451-82519328
传 真 0451-82519699
经 销 新华书店
印 刷 哈尔滨市石桥印务有限公司
开 本 787mm×1 092mm 1/16
印 张 13.75
字 数 343 千字
版 次 2011 年 3 月第 1 次版
印 次 2011 年 3 月第 1 次印刷
定 价 25.00 元
<http://press.hrbeu.edu.cn>
E-mail:heupress@hrbeu.edu.cn

前 言

C 语言是目前世界上最为流行、使用最为广泛的高级程序设计语言之一。它既具有高级语言的特点,又具有汇编语言的特点,因此,它的应用范围非常广泛,除了应用于软件开发外,在底层软件的开发中应用也非常普遍,比如单片机以及嵌入式系统开发。

“C 语言程序设计”课程是全国各大高校普遍设置的计算机基础课程之一,旨在培养学生计算机编程的基本思想、编程基本技能及逻辑思维能力。由于该课程涉及的语法和算法比较多,再加上通常授课的对象是大一的新生,他们普遍对计算机的工作原理不熟悉,因此在理解使用计算机解决问题的算法思路时容易出现问题,因而该课程普遍被学生认为比较难学。C 语言是一门实践性很强的课程,上机实践是学习 C 语言必不可少的一个重要环节,只有经过大量的编程训练,才能理解和学好 C 语言,因此,我们特地组织编写了这本《大学计算机(C 语言)实验及学习指导》。

本书包含 4 部分内容:第 1 部分详细介绍 C 语言程序设计上机实验的步骤以及 Turbo C ++ 3.0 运行环境,并就编程中常见的问题进行解析;安排 8 个实验,每个实验都给出了实验目的以及重点、难点,循序渐进地安排实验题目,从示例到习题,便于学生掌握每个章节的理论及编程技巧;第 2 部分为课程配套习题,针对教材的每个章节,安排了多种形式的习题,便于学生巩固知识点;第 3 部分为常见经典算法,旨在提高学生的编程能力和技巧;第 4 部分为综合模拟试题,可让学生检验自己的水平、训练学生的综合应用能力。

本书由高伟老师主编,丛晓红老师副主编,李江华教授主审。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、赵宝钢(按字母音序排序)等老师参加编写。

由于作者水平有限,编写时间仓促,书中难免存在错误和疏漏,敬请读者提出宝贵的意见和建议。

编 者

2011 年 2 月

目 录

第 1 章 程序设计(C 语言)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	10
1.3 选择结构程序设计实验	17
1.4 循环程序设计实验	27
1.5 函数实验	33
1.6 数组实验	41
1.7 结构体实验	49
1.8 指针实验	55
1.9 文件操作实验	62
第 2 章 程序设计同步练习指导	70
2.1 程序设计基础	70
2.2 数据表示及数据运算	70
2.3 最简单的 C 程序设计	75
2.4 选择结构程序设计	76
2.5 循环结构程序设计	84
2.6 函数	94
2.7 数组	106
2.8 常见算法	118
2.9 结构体	119
2.10 指针	121
2.11 文件	153
第 3 章 常用算法	155
3.1 程序设计基础	155
3.2 数据表示及数据运算	155
3.3 最简单的 C 程序设计	156
3.4 选择结构程序设计	157
3.5 循环结构程序设计	160
3.6 函数	166
3.7 数组	169
3.8 常见算法	179
3.9 结构体	180
3.10 指针	184
3.11 文件	189

第4章 考试样题	192
4.1 考试样题一	192
4.2 考试样题二	199
4.3 考试样题三	207

第1章 程序设计(C语言)实验指导

说明:本章的每一节都包含4个部分:验证实验、设计实验、自主研发实验和实测演练。验证实验是用来帮助对所学内容的理解;设计实验是在验证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;实测演练是用来测试基础知识的掌握情况和分析问题、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、验证实验

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。

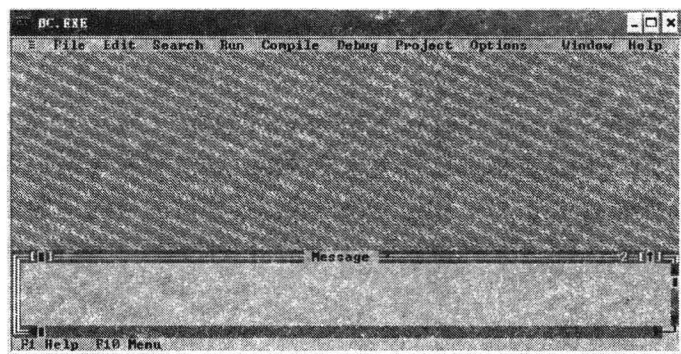


图 1.1 TC 程序窗口

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。

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

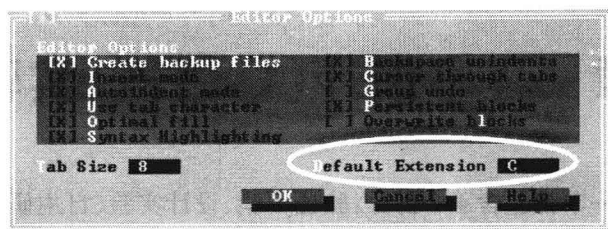


图 1.2 配置源程序默认扩展名

在编辑窗口的中间),如图 1.3 所示。

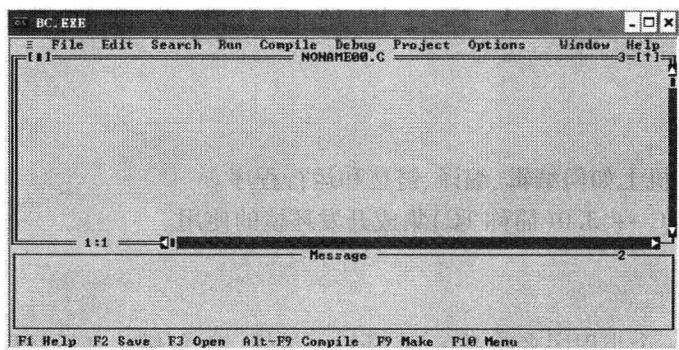


图 1.3 新建源文件窗口

在编辑窗口中输入下述程序:

```
#include <stdio.h>

void main()
{ printf("This is a C program. \n");
}
```

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。

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快捷键“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。

第 4 步:运行程序。

执行“Run”菜单的“Run”命令来运行程序,也可用快捷键“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。

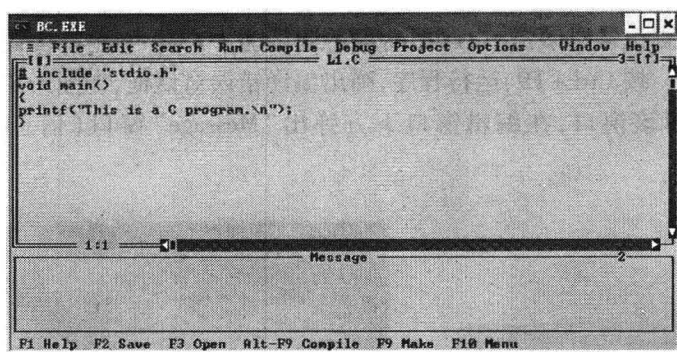


图 1.4 编辑窗口

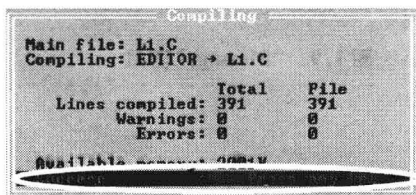


图 1.5 编译成功对话框

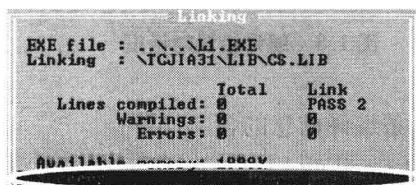


图 1.6 链接成功对话框

用“Windows”菜单的“User screen”命令来查看执行结果,也可用快捷键“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。

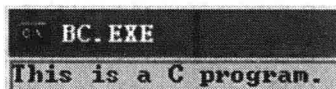


图 1.7 运行结果窗口

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“文件”菜单的“Quit”命名可关闭 TC 程序,也可用快捷键“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) 为主文件名命名时最好做到“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,如 \, /, <, >, |;
- (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、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。

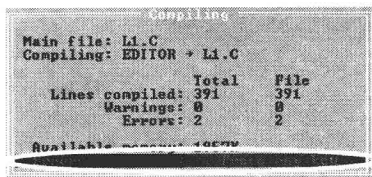


图 1.8 编译错误对话框

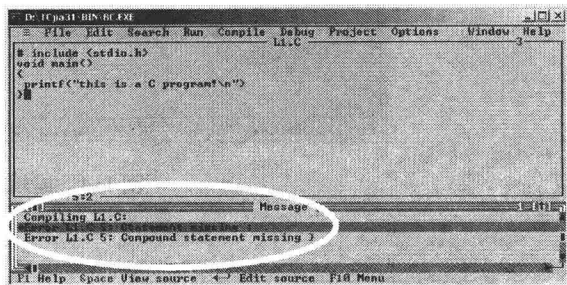


图 1.9 “Message”窗口

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,请注意查阅附录中的常见错误提示。

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。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,即第 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,表明该程序已通过了编译,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。

仿照上述操作方法,参考图 1.4,自行作几个验证性实验,举例如下:

- (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。

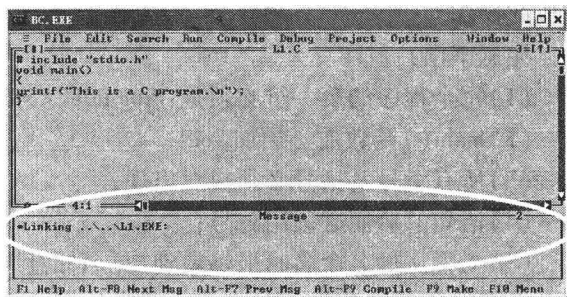


图 1.10 无编译及链接错误的状态

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。

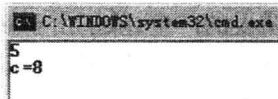


图1.11 运行结果

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。

4. 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。常用的几个快捷键如下:

- Ctrl + F9: 运行;
- Alt + F5: 查看结果;
- Alt + x: 退出 TC;
- Insert: 切换插入/改写状态;
- F1: 获取帮助;
- F6: 在各窗口之间进行切换;
- Alt + F3: 关闭窗口;
- Alt + 数字: 激活数字代表的窗口;
- F8: 单步执行程序;
- Ctrl + F7: 设置观察量;
- ESC: 返回上一菜单;
- Ctrl + Break: 运行时, 中断死循环状态, 返回到编辑状态。

(4) 在编译时会有三类常见的错误: 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, 编译立即停止, 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、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, 它只是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。

常见的编译信息如下。

- Statement missing ;: 语句缺少";"。编译程序发现一些表达语句后面没有分号。
- 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。
- 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。
- 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。
- 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 语句出现语法错误而引起。
- 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。
- 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。
- function call missing): 函数调用缺少")"。
- 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。
- division by zero: 除数为零。源文件的常量表达式中, 出现除数为零的情况。
- 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。

• 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。

• Illegal use of floating point:浮点运算非法。

• Redclaration of 'xxxxxxx': 'xxxxxxx'重定义。此标志符已经定义过。

• Subscripting missing]:下标缺少']'。编译程序发现一个下标表达式缺少右方括号。可能是由于漏掉或多写操作符或括号不匹配引起的。

• Too few parameters in call:函数调用参数不够。对带有原形的函数调用时,参数个数不够。

• Too many decimal points:十进制小数点太多。编译程序发现一个浮点常量中带有不止一个的十进制小数点。

三、设计实验

实验5:仿照实验1编写程序,在屏幕上输出如下内容:

```
*****
*           This is my first C program!           *
*****
```

实验6:编写程序,在屏幕上输出如下内容:

```
*****
* 100001, Bei Jing                                *
*                                                    *
*           Tian An Men You Ju                     *
*                                                    *
*                                                    *
*                               HRBEU, 150001        *
*****
```

实验7:编写程序,在屏幕上输出如下内容:

```
88888888
      88      88
    8  @@  @@  8
    8              8
      88   00   88
      88888888
```

实验8:仿照实验3编写程序,计算两个整数200和30的乘积。

实验9:仿照实验4编写程序,从键盘输入一个整数,计算该数与100的差。

四、自主研发实验

指导1:一般情况下程序由3个部分组成:输入部分、计算部分(或称为处理部分)及输出部分。根据实际问题的需要可以省略前两部分,但程序的输出部分是必不可少的。换句话说最简单的程序是由输出语句构成的,用输出的字母或符号可以构成图形或一些说明文字。编写程序给自己设计一张名片或打印一个卡通图片。

指导2:C表达式改变了传统的计算规则,初学时很容易出错。推导表达式的值时,首先作逻辑推理,然后再写一个小程序上机验证。例如,当 $a = 12$ 时,求表达式 $a += a - = a$

$\ast = a$ 的值。

指导3:通过输入语句增强了变量的表达能力,可以不修改程序而完成对多个量的计算。例如:从键盘给出半径的值,计算周长和面积,或者给出角度,计算对应的弧度。

五、实测演练

1. 填空题

(1)结构化程序设计的三种基本结构是_____、_____、_____。

(2)C 语句的结束符是_____。

(3)C 的程序有且仅有一个_____函数。

(4)C 函数的三个组成部分是_____、_____、_____。其中_____是必不可少的。

(5)将数学式 $\frac{1 + \frac{1}{2}b}{2c}$ 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形的,且 $\text{sub} = \text{pad} = 5$, $\text{pAd} = \text{sum}++$, $\text{pAd}++$, $++\text{pAd}$,则语句 $\text{printf}(\text{" \%d"}, \text{pad});$ 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)

(1)

```
# 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 porgram!          *\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、验证实验

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 10;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,并与前次运行结果作对比。

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

```