

CYUYAN CHENGXU SHEJI
SHIYANZHIDAO YU XITI JINGJIE

C语言程序设计

实验指导与习题精解

刘 锂 杨井荣 宋 容 主编



化学工业出版社

C语言程序设计

实验指导与习题精解

刘 锂 杨井荣 宋 容 主编
徐丽霞 肖 勇 副主编



化学工业出版社

· 北 京 ·

本书为更好地巩固和掌握 C 语言的相关知识,提高读者的程序阅读能力和程序设计能力而编写,是《C 语言程序设计教程》一书的配套实验与课程设计指导书。全书共有 8 个实验,每个实验均以全国计算机二级等级考试真题作为实例。书后附有教材习题的参考答案及解析。本书内容丰富,结构紧凑,选题典型丰富,对初学者具有很强的针对性。

本书可作为高等院校理工类专业计算机程序设计课程的教学用书,也可作为等级考试和自学人员的参考书。

图书在版编目(CIP)数据

C 语言程序设计实验指导与习题精解 / 刘锂, 杨井荣,
宋容主编. —北京: 化学工业出版社, 2018.3

ISBN 978-7-122-31401-7

I. ①C… II. ①刘… ②杨… ③宋… III. ①C 语
言-程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2018) 第 012773 号

责任编辑: 金 杰 王向军 徐一丹

文字编辑: 陈 喆

责任校对: 王 静

装帧设计: 韩 飞

出版发行: 化学工业出版社 (北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装: 三河市延风印装有限公司

787mm×1092mm 1/16 印张 7 字数 172 千字 2018 年 3 月北京第 1 版第 1 次印刷

购书咨询: 010-64518888 (传真: 010-64519686) 售后服务: 010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价: 26.00 元

版权所有 违者必究

前 言

C 语言是一门实践性很强的课程,要想真正学好 C 语言,必须通过大量的上机编程训练,在实践中逐步理解和掌握 C 语言程序设计的思路和方法。本书是《C 语言程序设计教程》一书配套使用的实验指导教材,也可独立作为上机实验和平时练习用书。

本书是集众多长期从事 C 语言教学工作的一线教师的经验和体会,并参考相关的国内外有关资料编写而成的。全书共有 8 个实验,每个实验都包含“实验目的及要求”“实验内容及指导”“思考与练习”。书后附有教材习题的参考答案及解析。本书内容丰富,结构紧凑,选题典型丰富,对初学者具有很强的针对性。

本书具有如下特点:

① 以掌握 C 语言基本语法规则、提高编程能力为目的,提供了题型丰富的练习题,包括程序填空题、程序改错题和程序设计题等。

② 知识结构组织合理,前后联系紧密。本书以实验内容及指导部分为实验重点,思考与练习部分便于读者完成实验后对所学知识进行练习和巩固。

③ 编程实验题目形式多样,实验指导方法灵活多变,有助于启发引导读者完成实验内容。注重实验内容的由浅入深和前后知识的连贯性,并把实验内容和计算机二级等级考试结合起来,使初学者能够进一步加深对课本相关内容的理解和掌握,更顺利地掌握 C 语言程序设计的方法和步骤,从而提高读者编程解决实际问题的能力。

④ 适合不同层次读者需求。本书难易搭配合理,读者在初学本课程时,等级考试知识部分可根据个人情况灵活选择。可以跳过等级考试知识部分直接学习其他内容以降低学习难度,后期复习时再进一步学习,也可以按教材顺序学习。

由于篇幅和课时的限制,在书中不可能介绍很多例子,只能介绍一些典型例题。建议读者除了完成教师指定的习题和实验外,尽可能阅读本书介绍的全部程序,并上机运行本书提供的全部实验内容以及自己感兴趣的程序,以开阔思路,提高编程能力。

本书由刘锂、杨井荣、宋容担任主编负责总体设计,徐丽霞、肖勇任副主编负责统稿及课件制作。分工为杨井荣(实验 1、习题答案及解析第 1 章)、宋容(实验 2、习题答案及解析第 2 章)、徐丽霞(实验 3、习题答案及解析第 3 章)、刘锂(实验 4、习题答案及解析第 4 章)、温爱红(实验 5、习题答案及解析第 5 章)、徐草草(实验 6、习题答案及解析第 6 章和第 9 章)、郭维树(实验 7、习题答案及解析第 7 章)、严南(实验 8、习题答案及解析第 8 章)。

在本书的编写过程中,笔者参阅了国内外诸多同行的著作,在此向他们致以衷心的感谢。

感谢读者选择使用本书,限于笔者学识水平,书中难免存在着不足之处,恳请读者批评指正。

编者
2017 年 12 月

目 录

第一部分 C 实验上机指导

实验 1 C 程序运行环境及基本操作	1
实验 2 程序的控制结构	8
实验 3 数组	13
实验 4 指针	20
实验 5 函数	26
实验 6 变量的存储类别与作用域	35
实验 7 用户自定义数据类型	39
实验 8 文件	43

第二部分 课后习题详解

习题 1 参考答案 (第 1 章 C 语言概述)	48
习题 2 参考答案 (第 2 章 程序的控制结构)	55
习题 3 参考答案 (第 3 章 数组)	59
习题 4 参考答案 (第 4 章 指针)	71
习题 5 参考答案 (第 5 章 函数)	75
习题 6 参考答案 (第 6 章 变量的存储类别与作用域)	82
习题 7 参考答案 (第 7 章 用户自定义数据类型)	84
习题 8 参考答案 (第 8 章 文件)	91
习题 9 参考答案 (第 9 章 公共基础知识)	93
附录 C 程序编译出错提示信息	97
参考文献	108

第一部分 C实验上机指导



实验1 C 程序运行环境及基本操作

一、实验目的及要求

- (1) 熟悉并掌握在 Visual C++ 6.0 环境中编辑、编译、连接、运行 C 语言程序的一般步骤。重点掌握 C 程序源文件、目标文件、可执行文件的建立和保存。
- (2) 进一步熟悉和掌握 C 语言源程序的基本结构及书写规则。
- (3) 初步熟悉各运算符的运算规则及使用。

二、实验内容及指导

【实验 1-1】在 Visual C++ 6.0（后文在不引起歧义的情况下简称 VC++）环境中编辑并运行以下程序 lab1-1.C。

```
#include <stdio.h>
void main()
{
    printf("Hello C!\n");
}
```



本题指导

(1) 本实验目的是练习并熟悉在 VC++ 环境中新建、保存源程序及进行编译、连接、运行的一般步骤。这里以英文版 Visual C++ 6.0 软件为例来说明。

(2) 本实验操作步骤如下：

① 在 Windows 中运行 VC++，进入集成开发环境。

② 新建源程序。在 VC++ 主窗口中运行“File”（“文件”）菜单上的“New”（“新建”）命令，如图 1-1 所示。

屏幕上弹出“New”对话框，单击此对话框上方的“Files”选项卡，选择“C++ Source File”项（即“C++源程序”），然后在“File”下的文本框中输入要新建的源程序文件名 lab1-1.C，如图 1-2 所示，单击“OK”按钮。

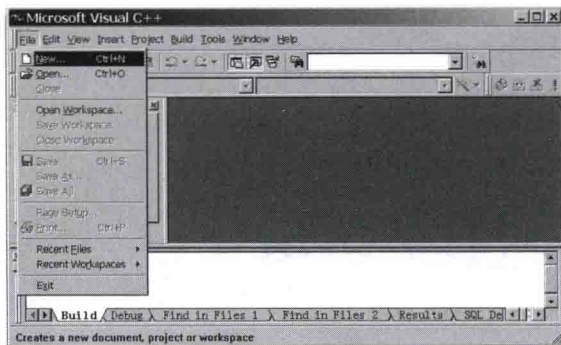


图 1-1 VC++窗口中的“File”菜单

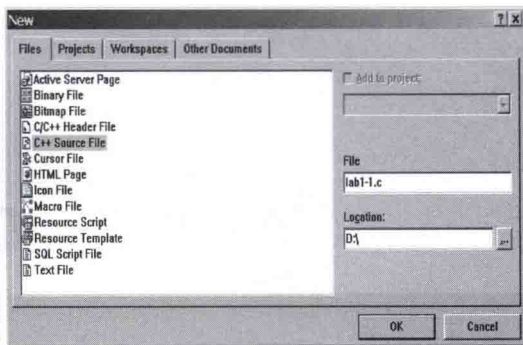


图 1-2 VC++窗口中的“New”对话框

单击“OK”按钮后回到 Visual C++主窗口，可以看到 VC++主窗口的标题栏显示出了源程序的文件名。

在主窗口的程序编辑窗口中输入源程序 lab1-1.C 的代码，如图 1-3 所示。

③ 保存源程序。输入完源程序之后，应执行“File”菜单中的“Save”（“保存”）或“Save as”（“另存为”）命令，保存程序（如果没有执行这一步操作，则执行下一步编译操作时，系统会自动提示是否保存源程序）。

④ 编译源程序。执行“Build”（“创建”）菜单的“Compile”（“编译”）命令，如图 1-4 所示。

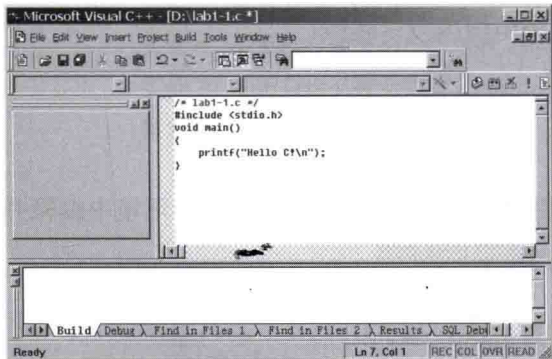


图 1-3 “lab1-1.C”程序编辑窗口

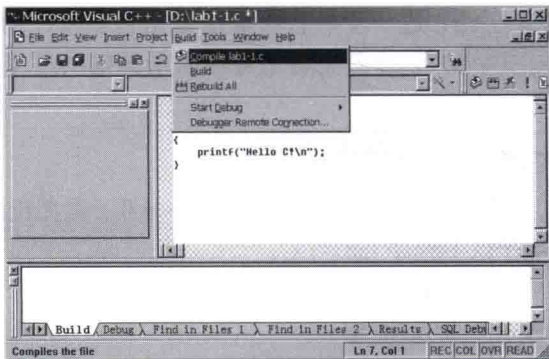


图 1-4 VC++窗口中的“Build”菜单

此时弹出如图 1-5 所示的对话框，询问是否要建立一个默认的工程工作区。单击“是”按钮，开始编译该文件。

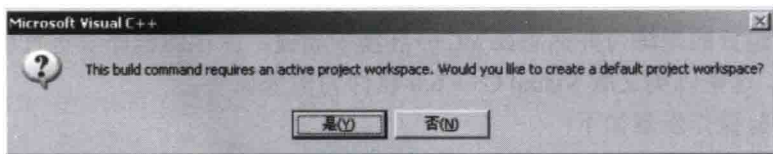


图 1-5 VC++对话框

编译过程中，如果源程序有错，则在 VC++主窗口下方的编译输出窗口的“Build”选项卡中显示出错，程序员须改正错误，编译改正才能完成。如果没有错，编译输出窗口就会显示生成的.OBJ 文件名，并显示如下信息：

lab1-1.OBJ - 0 error(s), 0 warning(s)

⑤ 连接。编译无误生成 OBJ 文件后, 就可执行“Build”菜单下的“Build lab1-1.exe”(“创建 lab1-1.exe”)命令或按“F7”键, 对被编译后的目标文件进行连接, 如图 1-6 所示。

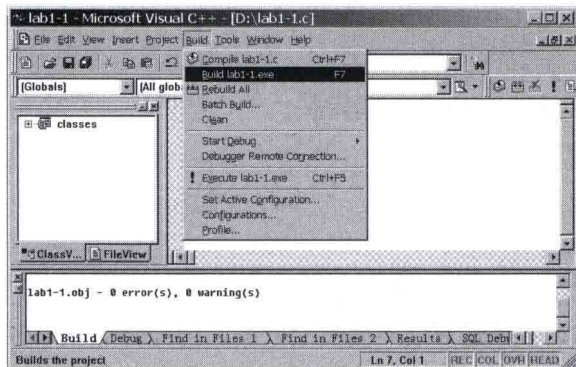


图 1-6 VC++ “Build” 菜单下的连接（创建）命令

如果连接过程中发现错误, 则会在编译输出窗口的“Build”选项卡中显示出连接错误的信息, 出现错误时, 修改源程序后再编译、连接, 直到连接没有错误为止。

编译连接成功时, 在编译输出窗口中就会显示如下信息:

lab1-1.exe - 0 error(s), 0 warning(s)

表明没有错误和警告信息, 创建了可执行文件 lab1-1.exe。

⑥ 运行程序。连接生成 EXE 文件后, 就可执行“Build”菜单下的“Execute”(“执行”)命令或按“Ctrl”+“F5”键(如图 1-7 所示), 来运行程序 lab1-1.exe。程序执行时将弹出一个 DOS 窗口并输出以下内容:

Hello C!

Press any key to continue (按任意键继续)

按任意键后, 输出结果窗口将关闭, 回到源程序编辑窗口。

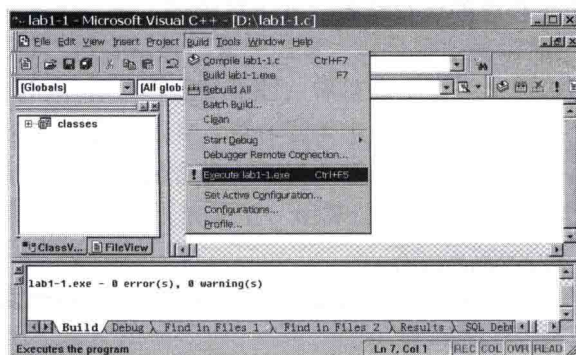


图 1-7 VC++ “Build” 菜单下的 Execute 命令

【实验 1-2】 分析以下程序的运行结果, 并上机验证。

```
#define M 6
#include "stdio.h"
void main()
{
```



```

char c='k';
int a,b;
int i=1,j=2,k=3;
a=b=8;
printf("%d,%d\n",++a,b--);
printf("%d,%d\n",'a'+5<c,1<M<j);
printf("%d,%d\n",i+j+k==2*j,k==j==i+5);
}

```



本题指导

本题虽只有短短几行，却涵盖了宏定义、头文件包含、变量定义及初始化、赋值运算、自增/减运算、算术运算、关系运算及不同类型数据间的混合运算等多个内容，值得仔细分析和体会。

(1) 注意自增运算符“++”和自减运算符“--”放在变量前面与后面时运算顺序的区别（放在前面是先自增/自减，然后取变量的值；放在后面则是先取变量的值，后进行自增/自减运算）。

(2) 程序第一行是一个无参数的宏定义，在较早的 C 软件版本中称为符号常量定义。

(3) 注意语句“a=b=8;”的执行过程。该语句相当于“a=(b=8);”，先执行右边括号内的赋值操作，括号内表达式的值为 8，再给变量 a 赋值。

(4) 注意表达式 $1 < M < j$ 的运算过程。不能理解为数学上的两个不等式，而是相当于 $(1 < M) < j$ ，整个是一个关系不等式，括号内也是一个关系表达式。求值时先求括号内表达式的值，由于 $1 < M$ 为真，因此整个表达式相当于 $1 < j$ ，该关系表达式也为真，因此整个表达式的值为 1。

(5) 根据算术运算符和关系运算符的优先级，最后一行中的两个表达式分别相当于 $(i+j+k) == (-2*j)$ 和 $(k==j) == (i+5)$ ，读者不难求出其值。

【实验 1-3】 分析以下程序的运行结果，并上机验证。

```

#include <stdio.h>
void main()
{
    int x, y, z;
    x=y=z=0;
    ++x||++y&&++z;
    printf("x=%d,y=%d,z=%d\n",x,y,z);
    x=y=z=0;
    ++x&&++y||++z;
    printf("x=%d,y=%d,z=%d\n",x,y,z);
    x=y=z=-1;
    ++x&&++y||++z;
    x=y=z=2;
    printf("%d,%d,%d\n",!x*y,!!x,x==5&&y&&(z=8));
}

```



本题指导

本题主要让读者加深对逻辑运算符及逻辑表达式求解过程的理解。

(1) “||” 运算符左边表达式的值为真(非 0)时, 不再计算右边的表达式。

(2) “&&” 运算符左边表达式的值为假(0)时, 不再计算右边的表达式。

【实验 1-4】 运行以下程序, 分析各表达式输出的结果。

```
#include<stdio.h>

void main()
{int a,b,c,i,j;
 printf("%d\n",5%3*2/6-!5);
 printf("%d\n",5>=2+1-(3<5));
 i=2;
 j=4;
 printf("%d\n",++i-j--);
 printf("%d\n",i+=i*=j);
 printf("%d\n",i=3/2*(j=3-2));
 a=1;
 b=2;
 c=3;
 printf("%d\n",!a&&b++);
 printf("%d\n",a||b+2&&a*b);
 printf("%d\n",++a,b=10,a+5);
 printf("%d\n",a+=b%=a+b);
 printf("a=%d,b=%d,c=%d\n",a,b,c);
 printf("%d\n",a>b?a:c>b?c:b);
}
```



本题指导

本题涉及算术、逻辑、符合赋值、条件运算、逗号等多重运算符和表达式。

(1) 在分析表达式的运算过程及结果时, 需注意掌握并分析以下几个方面:

- ① 表达式中各种运算符的功能和对操作数类型的要求;
- ② 根据各运算符的优先级和结合性, 正确分析运算的先后次序;
- ③ 根据各运算符的功能、操作数类型、运算过程中的类型转换, 确定运算结果的值及类型。

(2) 逗号表达式求解时, 会顺序执行每一个表达式, 整个表达式的值为最后一个表达式的值。

(3) 对于赋值和复合赋值运算, 需注意以下两点:

① 复合赋值运算的过程是, 首先根据赋值运算符前的运算符, 复合赋值运算符左边的变量和复合赋值运算符右边的表达式进行相应的运算, 然后将该运算结果赋值给复合赋值运算符左边的变量。例如, 表达式 “ $x+=y+1$ ” 等于 “ $x=x+(y+1)$ ”。

② 对于赋值和复合赋值表达式, 其结果类型为赋值和复合赋值运算符左边变量的类型。

【实验 1-5】 按下面要求运行程序 lab1-5.C, 分析并观察运行结果。理解字符的输入与输

出, 对结果进行合理的解释。

```
#include <stdio.h>

void main()
{
    int a,b,c;
    char x,y,z;
    scanf("a=%d,b=%d",&a,&b);
    scanf("%c%c%c",&x,&y,&z);
    c=a+b;
    printf("%d+%d=%d",a,b,c);
    printf("%c%c%c",x,y,z);
}
```



本题指导

本题主要是练习字符数据的输入输出, 并通过练习了解 `getchar`、`putchar`、`scanf` 和 `printf` 等函数在处理字符数据时的用法。

(1) 利用 `scanf` 和 `printf` 函数实现外设和内存的数据交互, 实现数据的标准输入输出。

(2) 利用 `getchar` 和 `putchar` 函数可以实现一个字符数据的输入输出。调用 `putchar` 函数的时候, 直接把待输出的字符常量或者字符变量放到参数的圆括号内, 例如输出字符 'a'; 那么可以写成 `putchar('a');`, 每调用一次该函数, 可以输出一个字符。调用 `getchar` 函数的时候, 直接从键盘输入一个字符赋值给该函数左侧的变量。输入输出一个字符时, 可以用 `scanf` 和 `printf` 函数, 也可以用 `getchar` 和 `putchar` 函数。

【实验 1-6】 分析以下程序的输出结果并上机验证。

```
#include<stdio.h>
void main()
{
    short int a=52767,b=a+1;
    char c1=97,c2='b';
    unsigned int d=54321;
    unsigned long e=1234567;
    float f=1.56789,g=-345.48;
    double m=12.67323,n=123.456789;
    printf("%d,%d\n",a,b);
    printf("%#o,%#lx\n",d,e);
    putchar(c1);
    printf("%c %c\n",c1,c2);
    printf("%+6f,%+5.4f\n",f,g);
    printf("m=%lf,n=%e\n",m,n);
}
```



本题指导

本题主要是练习调用库函数 `printf` 来输出各种类型的数据。

(1) 关于 `printf` 函数的几点说明:

① printf 函数的一般格式为:

printf(格式控制串,输出列表);

其中,格式控制串是用双引号引起来的字符串,也称转换控制字符串;输出列表是需要输出的一些数据项列表,数据项可以是变量或任何表达式。

② 格式控制串中的字符分为两类,一类是普通的 ASCII 码字符(包括转义字符),这些字符在输出时将原样输出到屏幕;另一类是以百分号(%)和其他字符组成的格式控制符,如果输出正常的话,在输出时格式控制符将被输出列表中相应位置上数据项的值替代。例如,在

```
printf("x=%d,y=%d,x+y=%d\n",x,y,x+y);
```

中,格式控制串中的字符'x'、'y'、'+'、'='、'\n'等都是普通 ASCII 码字符,而%d是格式控制符。

③ 在 printf 函数中,格式说明符与输出项从左到右在类型上必须一一对应匹配。如不匹配,将导致数据不能正确输出,但此时 C 系统不报错。另外,格式说明符与输出项的个数一般应该相同,如果格式说明的个数少于输出项的个数,多余的输出项不予输出;如果格式说明的个数多于输出项的个数,则对于多余的格式将输出不定的值。

有关 printf 函数更详细的用法,请参考理论教材或其他资料。

(2) 在给变量进行初始化或赋值时,一定要注意变量所表示的数的范围。如本例中整数 32767 加 1 后输出为-32768。

【实验 1-7】输入并运行下面的程序。为使变量 a、b、x、y、c1、c2 的值分别为 3、7、3.2、2.4、'a'、'b',运行程序时正确的数据输入形式是什么?

```
#include<stdio.h>
void main()
{
    int a,b;
    float x,y;
    char c1,c2;
    scanf("a=%d b=%d",&a,&b);
    scanf("x=%f y=%f",&x,&y);
    scanf("c1=%c c2=%c",&c1,&c2);
    printf("a=%d,b=%d,x=%f,y=%f,c1=%c,c2=%c\n",a,b,x,y,c1,c2);
}
```



本题指导

(1) 本题进一步练习 scanf 函数及其对应的键盘输入格式。

scanf 函数的一般形式可写为:

scanf(格式控制串,输入项地址列表);

其中格式控制串中的字符分为两类,一类是普通的 ASCII 码字符(包括部分转义字符如'\n'、'\t'等),这些字符在程序运行时必须从键盘原样输入;另一类是以百分号(%)和其他字符组成的格式控制符,输入时格式控制符位置上应输入对应类型的数据值常量。输入项地址列表是变量或数组元素的地址。

(2) 说明:如果要获得以上的数据,正确的输入应该是(" "表示空格符):


```
a=3 b=7 x=3.2 y=2.4 c1=a c2=b ✓
```

由于输入时 `scanf` 函数若遇到回车键将视为输入结束，因此，如果在输入完 “`a=3 b=7`” 后直接回车，则后面的几个变量则无法正确接收数据。

(3) 由于 `scanf` 函数的格式控制串中，普通 ASCII 码字符（包括部分转义字符如 ‘\n’、‘\t’ 等）应原样输入，为了避免输入时出错或过于烦琐，在格式控制串中应尽量少使用不必要的普通字符。

三、思考与练习

(1) C 语言程序的书写需要注意哪些方面？

提示：最基本的方面包括两方面。一是保证程序书写的正确性；二是要有良好的书写风格（如格式清晰、添加必要的注释等），以便自己和他人容易阅读。

(2) 通过查阅本书附录 II 或其他资料，熟悉编译、连接过程中常用提示信息含义。

提示：正确理解这些提示信息，便于查找及改正源程序中的错误和其他错误。有经验的程序员会非常注意这些提示信息。

(3) 分析下面程序的运行结果，并上机验证。

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

(4) 在 `printf` 函数中，如果输出项列表与格式符的数目不匹配，会有怎样的输出结果？同样地，如果是在 `scanf` 函数中出现了输入项地址列表与格式符不相匹配的情况，又会对输入造成怎样的影响？试编写程序验证。

实验 2 程序的控制结构

一、实验目的及要求

- (1) 熟悉并掌握 `if` 语句、`switch` 语句的书写格式、功能及执行过程。
- (2) 熟悉并掌握 `if` 语句、`switch` 语句的嵌套。
- (3) 掌握 `break` 语句在 `switch` 语句中的功能和用法。
- (4) 熟悉并掌握 `while` 语句、`do-while` 语句、`for` 语句的格式及用法。
- (5) 熟悉并掌握较简单的循环嵌套。
- (6) 熟悉并掌握 `continue` 和 `break` 语句的功能和用法。

二、实验内容及指导

(1) 程序填空题。

【实验 2-1】下列给定程序的功能是计算如下公式。

$$s = \frac{3}{2^2} - \frac{5}{4^2} + \frac{7}{6^2} - \cdots (-1)^{n-1} \frac{(2n+1)}{(2n)^2} \text{ 直到 } \left| \frac{(2n+1)}{(2n)^2} \right| \leq 10^{-3}, \text{ 并且把计算结果输出。}$$

例如, 若 e 的值为 $1e-3$, 输出结果为 0.551690 。请在下画线处填入正确的内容并将下画线删除, 使程序得出正确的结果。

注意: 不得增行或删行, 也不得更改程序的结构!



实验指导

- ① 基本算术运算时的类型转换。
- ② 如何通过循环变量变化实现循环操作。

```
#include <stdio.h>
main()
{ double e=1e-3;
  int i, k; double s, t, x;
  s=0; k=1; i=2;
  /*****found*****/
  x=___1___/4;
  /*****found*****/
  while(x ___2___ e)
  { s=s+k*x;
    k=k*(-1);
    t=2*i;
  /*****found*****/
    x=___3___/(t*t);
    i++;
  }
  printf("\nThe result is: %f\n",s);
}
```

【实验 2-2】下列给定程序的功能是: 输入一个整数 n , 并把 n 中各位上为偶数的数取出, 并按原来从高位到低位相反的顺序组成一个新数输出。

例如, 输入一个整数 27638496 , 函数返回值为 64862 。

请在下画线处填入正确的内容并将下画线删除, 使程序得出正确的结果。

注意: 不得增行或删行, 也不得更改程序的结构!



实验指导

如果知道某数 n 的各个位的数值, 可以得到该数值 n , 如 n 的个位为 a 、十位为 b 、百位



为 c , 那么 $n=c \times 100+b \times 10+a$ 。如果知道数值 n , 可以采用求余和除法操作来表示其各个位, 如 $n \% 10$ (取个位)、 $n / 10$ (取十位), 读者可以自己考虑一下, 如果是三位数, 如何提取各个位? 四位数呢?

```
#include <stdio.h>
main()
{ unsigned long n=-1,x=0;
  int t;
  while(n>99999999||n<0)
  { printf("Please input(0<n<100000000): "); scanf("%ld",&n); }
  while(n)
  { t=n%10;
    /******found******/
    if(t%2==__1__)
    /******found******/
    x=__2__+t;
    /******found******/
    n=__3__;
  }
  printf("\nThe result is: %ld\n",x);
}
```

(2) 程序调试题。

【实验2-3】下列给定程序的功能是: 将长整型数中各位上为奇数的数依次取出, 构成一个新数放在 t 中。高位仍在高位, 低位仍在低位。

例如, 当 s 中的数为 87653142 时, t 中的数为 7531。

请改正程序中的错误, 使它能得出正确的结果。

注意: 不得增行或删行, 也不得更改程序的结构!



实验指导

- ① 初始值与后面的计算表达式紧密相连。
- ② if 语句条件表达式, 结合奇偶数的表示方法来确定该表达式内容。

```
#include <stdio.h>
main()
{ long s, t,d,sl=1;
  printf("\nPlease enter s:"); scanf("%ld", &s);

  /******found******/

  t = 1;
  while ( s > 0)
  { d = s%10;
    /******found******/
    if (d%2 == 0)
```



```

    { t = d * sl + t;
sl *= 10;
    }
    s /= 10;
}
printf("The result is: %ld\n", t);
}

```

(3) 编程题。

【实验 2-4】 试用 switch 语句实现把百分制转换成等级制，要求在输入一个百分制分数后能输出对应的等级。对应关系如下：

百分制分数	五分制分数	百分制分数	五分制分数
≥ 90	A	≥ 60	D
≥ 80	B	< 60	E
≥ 70	C		



实验指导

① 本题的目的是练习使用 switch 语句及 scanf、printf 函数。对于 switch 语句，主要是掌握其书写格式、功能和执行过程。

② switch 后面的表达式要求是整型表达式或字符型表达式，对应地，case 语句中的常量值应是整型或者是字符类型。“case 常量:”实际上相当于一个语句标号，是多分支的入口。

③ switch 语句在执行时，首先计算常量表达式的值，然后根据常量表达式的值从匹配的某个 case 语句或 default 语句处开始执行，执行过程中遇到 break 语句则会结束当前 switch 语句；否则将一直往后执行，直到其语句体内的所有语句全部执行完毕。

④ 本题参考算法如下：

/*实验 2-4 程序算法*/

begin

按格式从键盘输入数 n

switch(取数 n 的十位)

{

case 10:

case 9:

输出结果 A; break;

case 8:

输出结果 B; break;

case 7:

输出结果 C; break;

case 6:

输出结果 D; break;



```
case 5:
case 4:
case 3:
case 2:
case 1:
case 0:
    输出结果 E; break;
}
end
```

【实验 2-5】输入数字 n ，用 while 循环求出 $n!$ 的值并输出。



实验指导

① 阶乘的运算公式是： $n!=1\times 2\times \cdots \times (n-2)\times (n-1)\times n$ ，($n>0, n\in N$)。因此在程序中可通过 while 循环实现累乘功能。

② n 必须为正整数，所以 n 可以定义为 int 型，在输入 n 的值之后要判断它是否大于等于 0。当 n 较大时，阶乘值 m 的值可能较大，因此最好将 m 定义为长整型。

程序算法如下：

```
/*实验 2-5 程序算法*/
begin
    i=1,m=1
    输入正整数 n
    if(n>=0)
        while(i<=n)
        {
            m=m*i
            i++
        }
    else
        输出错误的提示信息
end
```

【实验 2-6】猴子吃桃问题。猴子第一天摘下若干个桃子，当即吃了一半，不过瘾，又多吃了一个。第二天又将剩下的吃了一半，又多吃了一个，以后每天都吃剩下的一半多一个，到第 10 天时，只剩下一个桃子。求猴子第一天共摘了多少个桃子。



实验指导

- ① 根据题意得知，第 10 天： $p_{10}=1$
第 9 天： $p_9/2-1=p_{10}$
 $p_9=(p_{10}+1)\times 2$
第 8 天： $p_8/2-1=p_9$
 $p_8=(p_9+1)\times 2$