

第一篇 基础知识模块

基础知识模块是中等职业学校计算机及应用专业学生必须掌握的知识，其教学目标是：

1. 理解编程语言的基本概念，具有使用 C 语言编程的能力；
2. 掌握结构化程序设计的方法和技巧；
3. 掌握 C 语言的基本语法、基本符号、词汇等；
4. 掌握数据类型、函数、语句的基本知识及其应用；
5. 理解编程语言有关算法的思想；
6. 掌握数组知识和使用方法；
7. 初步掌握文件的使用方法；
8. 具有阅读程序的能力并掌握上机调试程序的方法。

同时在教学中，注意锻炼学生的逻辑思维能力；教育学生具有严谨的学风、创新意识和创新精神、科学的求学态度，以及互助合作的团队精神。

第 1 章 C 语言概述



C 语言是编程语言中较为流行的一种。随着计算机的普及和发展，C 语言在各个领域的应用越来越广泛。几乎各类计算机都支持 C 语言的开发环境，这对 C 语言的普及及应用奠定了基础。本章主要介绍 C 程序的结构、C 语言的基本语法、C 语言集成开发环境等 C 语言的基本知识。

【本章要点】

1. C 语言的发展及特点；
2. C 程序的基本结构；
3. C 语言的基本符号与词汇；
4. C 语言集成开发环境。

【学习目标】

1. 掌握 C 程序的基本结构；
2. 掌握 C 语言的基本符号与词汇；
3. 了解 Turbo C 集成开发环境的基本使用方法；
4. 熟悉在 Turbo C 中编辑和运行最简单的 C 程序。

【课时建议】

讲授 2 课时，上机 2 课时（利用机动课时）。

1.1 C 语言简史及特点

1.1.1 C 语言的发展

C 语言是一种编译性程序设计语言，它与 UNIX 操作系统紧密地联系在一起。UNIX 系统是通用的、交互式的计算机操作系统，它诞生于 1969 年，是由美国贝尔实验室的 K.Thompson 和 D.M.Ritchie 用汇编语言开发成功的。

C 语言的前身是 BCPL 语言。1967 年英国剑桥大学的 Martin Richard 推出 BCPL 语言(Basic Combined Programming Language)。1970 年贝尔实验室的 K.Thompson 以 BCPL 语言为基础，开发了 B 语言，并用 B 语言编写了 UNIX 操作系统，在 PDP-7 计算机上实现。1972 年贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出 C 语言，C 语言既保持了 BCPL 语言和 B 语言的精练、接近硬件的优点，又克服了它们过于简单的缺点。1973 年，K.Thompson 和 D.M.Ritchie 合作把 UNIX 的 90%以上用 C 语言改写，并加进了多道程序设计的功能，称为 UNIX 第五版，开创了 UNIX 系统发展的新局面。1975 年 UNIX 第六版颁布后，C 语言得到计算机界的普遍认可，从此，C 语言与 UNIX 系统一起互相促进，迅速发展。

最初，设计 C 语言的目的只是为了描述和实现 UNIX 操作系统。目前，C 语言已独立于 UNIX

系统,先后被移植到大、中、小型计算机及微机上。1978 年 B.Kernighan 和 D.M.Ritchie 合作编写了经典著作“ The C Programming Language”,它是目前所有 C 语言版本的基础。1983 年美国国家标准化协会(ANSI)对 C 语言问世以来的各种版本进行了扩充,制定了 ANSI C。现在流行的 C 语言版本有:Microsoft C, Turbo C, Quick C, Borland C 等。本书主要介绍 Turbo C。

1.1.2 C 语言的特点

C 语言具有以下几个基本特点:

(1) C 语言是结构化程序设计语言。C 语言程序的逻辑结构可以用顺序、选择和循环三种基本结构组成,便于采用自顶向下、逐步细化的结构化程序设计技术。用 C 语言编制的程序,具有容易理解、便于维护的优点。

(2) C 语言是模块化程序设计语言。C 语言的函数结构、程序模块间的相互调用及数据传递和数据共享技术,为大型软件设计的模块化分解技术,以及软件工程技术的应用提供了强有力的支持。

(3) C 语言具有丰富的运算能力。C 语言除了具有一般高级语言所拥有的四则运算及逻辑运算功能外,还具有二进制的位(bit)运算、单项运算和复合运算等功能。

(4) C 语言具有丰富的数据类型和较强的数据处理能力。C 语言不但具有整型、实型、双精度型,还具有结构、联合等构造类型,并为用户提供了自定义数据类型。此外,C 语言还具有预处理能力,能够对字符串或特定参数进行宏定义。

(5) C 语言具有较强的移植性。C 语言程序本身并不依赖于计算机的硬件系统,只要在不同种类的计算机上配置 C 语言编译系统,即可达到程序移植的目的。

(6) C 语言不但具有高级语言的特性,还具有汇编语言的特点。C 语言既有高级语言面向用户、容易记忆、便于阅读和书写的优点;又有面向硬件和系统,可以直接访问硬件的功能。

(7) C 语言具有较好的通用性。它既可用于编写操作系统、编译程序等系统软件,也可用于编写各种应用软件。

1.2 C 语言程序

1.2.1 几个典型的 C 程序

C 语言的源程序由一个或多个函数组成,每个函数完成一种指定的操作,所以有人又把 C 语言称为函数式语言。下面,通过 3 个简单的例子来了解 C 程序的基本结构。

【例 1.1】运行时在屏幕上显示信息“Hello!”。

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

运行结果:

```
Hello!
```

程序说明:

(1) C 程序由一系列函数组成,这些函数中必须有且只能有一个名为 main 的函数,这个函数称为主函数,整个程序从主函数开始执行。在【例 1.1】中,只有一个主函数而无其他函数。

(2) 程序第一行中的 main 是主函数的函数名,main 后面的一对小括号是函数定义的标志,



不能省略。

(3) 程序第三行的 printf 是 C 语言的格式输出函数, 在本程序中, printf 函数的作用是输出括号内双引号之间的字符串, 其中 “\n” 代表换行符。第三行末尾的分号, 则是 C 语句结束的标志。

(4) 程序第二行和第四行是一对大括弧, 在这里表示函数体的开始和结束。一个函数中要执行的语句都写在函数体中。

【例 1.2】 求两个数之和。

```
main()
{
    int a,b,sum;    /* 定义三个变量 */
    a=10;b=26;    /* 分别给变量 a 和 b 赋值 */
    sum=a+b;      /* 求 a 和 b 的和, 并把结果放入变量 sum 中 */
    printf("sum=%d\n",sum);    /* 输出变量 sum 的值 */
}
```

运行结果:

```
sum=36
```

程序说明:

(1) 这个程序由一个主函数组成, 其中, 第三行的 int 表示定义变量类型为整型, 该行定义了 a、b、sum 3 个整型变量。

(2) 程序第四行和第五行中的语句均为赋值语句, “=” 为赋值运算符, 其作用是将其右边的常量或表达式值赋值给左边的变量。

(3) 第六行中的 “%d” 是输入/输出函数中的格式字符串, 在此表示以十进制整数的形式输出变量 sum 的值。程序的运行结果中, “%d” 的位置被 sum 的值取代。

(4) 程序中多次出现的 “/*” 和 “*/” 是一对注释符, 注释的内容写在这对注释符之间。注释内容对程序的编译和运行不起任何作用, 其目的是为了提高程序的可读性。在必要的地方给程序加上注释是一个好习惯, 这使得程序易于理解, 而对程序的理解是进一步修改和调试程序的基础。

【例 1.3】 求一个数的平方。

```
main()
{
    int a,p;        /* 定义两个整型变量 */
    scanf("%d",&a); /* 输入一个整数到变量 a 中 */
    p=f(a);        /* 调用 f 函数求 a 的平方, 并把函数的返回值赋值给变量 p */
    printf("%d",p); /* 输出变量 p 的值 */
}
f(n)              /* 定义 f 函数, n 为形式参数 */
int n;            /* 定义形式参数的类型 */
{
    int t;
    t=n*n;        /* 求 n 的平方, 并把结果赋值给变量 t */
    return t;     /* 返回变量 t 的值 */
}
```

运行结果:

```
5
```

25

程序说明：

(1) 这个程序的功能是，输入一个整数，输出该数的平方值。程序由两个函数组成，一个是 main 函数，一个是 f 函数。这两个函数的定义是相互独立的。

(2) f 函数的功能是求 n 的平方，并返回 n 的平方值。main 函数的第五行调用 f 函数时，把变量 a 的值传递给形式参数 n，因此，调用 f 函数的结果是求得了 a 的平方。

1.2.2 C 程序的基本结构

通过上面 3 个例子，可以把 C 程序的基本结构归结如下：

(1) C 语言程序由函数构成。函数是构成 C 程序的基本单位，即 C 程序由一个或多个函数组成，其中必须有且只能有一个名为 main 的主函数。如在前面的【例 1.1】和【例 1.2】中，均只有一个 main 函数，而在【例 1.3】中，则有 main 和 f 两个函数。

(2) 每个函数的基本结构如下：

```
函数名()
{
    语句 1;
    .....
    语句 n;
}
```

有的函数定义时，函数名后的小括号内有形式参数，如【例 1.3】中的 f 函数。{} 内则是由若干语句组成的函数体，每个语句必须以分号结束。C 语言的书写格式较自由，一行内可以写多个语句（如【例 1.2】程序的第四行有两个赋值语句），一个语句很长时也可以分写在多行上。

(3) 各个函数的定义是相互独立的。各函数定义的顺序无关紧要，主函数可以定义在其他函数之前，也可以定义在其他函数之后，但程序的执行总是从主函数开始。

1.2.3 C 语言的基本符号与词汇

任何程序设计语言都规定了自己的一套基本符号和词汇，C 语言也不例外。

1. C 语言的基本符号集

C 语言的基本符号集采用 ASCII 码字符集，包括：

(1) 大小写英文字母各 26 个。

(2) 10 个阿拉伯数字 0~9。

(3) 其他特殊符号，包括以下运算符和操作符：

+	-	*	/	%	<
<=	>	>=	==	!=	&&
	!	&		~	=
++	--	?:	<<	>>	()
[]	.	->	^	#	sizeof
+=	-=	*=	/=	%=	&=
^=	=	,			

2. C 语言的词汇

(1) 标识符。程序中用来标识变量名、函数名、数组名、数据类型名等的有效字符序列称为标识符。简单地说，标识符就是一个名字。

标识符的构成规则如下：

标识符只能由英文字母 (A~Z, a~z)、数字 (0~9) 和下划线 (_) 三类符号组成，但第一字符必须是字母或下划线。例如，下面的标识符是合法的：

```
sum, Sum, n2, _average, a_3, student_2_name
```

下面是不合法的标识符：

```
num-1, a#3, 2student, !sum_2, number.3
```

大写字母与小写字母含义不同。如 sum、Sum、SUM 表示 3 个完全不同的标识符。

一般的 C 编译系统只取标识符的前 8 个字符为有效字符，而 Turbo C 则取标识符的前 32 个字符为有效字符。

通常，命名标识符时应该做到“常用取简，专用取繁”。一般情况下，标识符的长度在 8 个字符以内就可以了。

(2) 关键字。关键字又称为保留字，是 C 语言编译系统所固有的、具有专门意义的标识符。C 语言的关键字有 32 个，一般用作 C 语言的数据类型名或语句名，如表 1-1 所示。

表 1-1 C 语言关键字

描述类型定义	描述存储类型	描述数据类型	描述语句
Typedef	auto	char	break
Void	extern	double	continue
	static	float	switch
	register	int	case
		long	default
		short	if
		struct	else
		union	do
		unsigned	for
		const	while
		enum	goto
		signed	sizeof
		volatile	return

说明：

所有关键字的字母均采用小写。

关键字不能再作为用户的常量、变量、函数和类型等的名字。

1.3 C 语言集成开发环境

1.3.1 Turbo C 集成开发环境介绍

C 语言有许多种编译器, 这些编译器之间只有很小的区别, 只要学会其中的一种, 对其他几种就能很快的适应。本节主要介绍在微机较为流行的 C 语言版本——Turbo C。

Turbo C 是美国 Borland 公司的产品。它是一个集程序建立、编辑、编译、连接、运行于一体的集成开发环境, 是一个易学、易用、快速高效、方便用户使用的程序开发工具。Turbo C 具有图形库和文本窗口函数库, 具有查错及协处理器的仿真功能。当前在国内使用比较普遍的是 Turbo C 2.0, 它是在 MS-DOS 操作系统上使用的编译系统, 也可以在 Windows 操作系统下运行。本书的所有例题都是在该环境下调试通过的, 后面的介绍都以此为基础。

1.3.2 Turbo C 集成开发环境的使用

1. 启动 Turbo C

假设 Turbo C 安装在 C 盘根目录下的 TC 文件夹中, 那么在 Windows XP 下运行 Turbo C 的步骤如下:

(1) 单击屏幕左下角的“开始”按钮, 选择“所有程序”/“附件”/“命令提示符”, 打开“命令提示符”窗口, 如图 1.1 所示。



图 1.1 “命令提示符”窗口

(2) 在提示符后输入“cd\”, 按回车键后回到 C 盘根目录下。

(3) 在提示符后依次输入以下命令即可启动 Turbo C:

```
C:\>cd tc
```

```
C:\TC>tc
```

Turbo C 启动后, 屏幕上即显示出其开始界面, 如图 1.2 所示。

2. C 源程序的建立和编辑

(1) 源程序的建立。在 Turbo C 中新建一个 C 源程序时, 可以直接选择“Edit”(编辑)菜单, 当光标停留在编辑窗口中时, 就可以输入程序了。每输完一行后, 按回车键将光标移到下一行的开始位置。

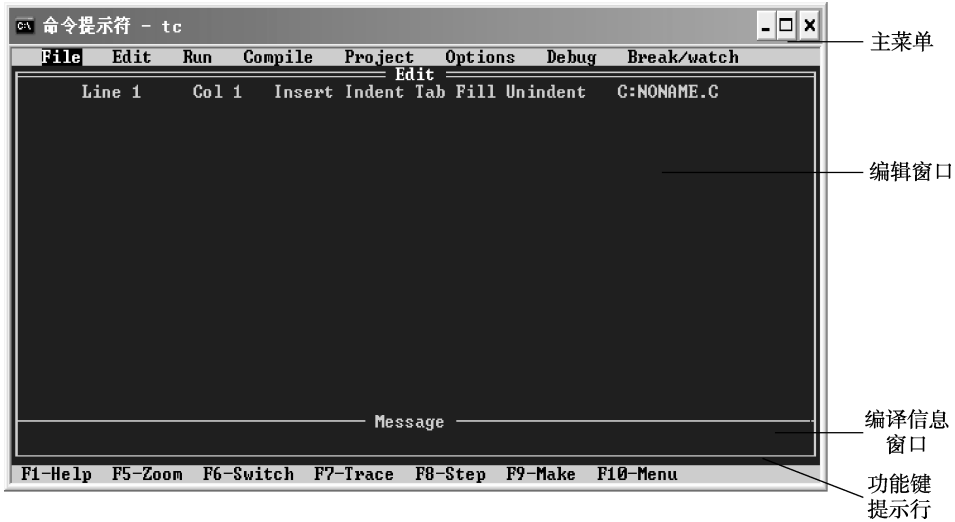


图 1.2 Turbo C 2.0 的开始界面

这里，在编辑窗口中输入【例 1.1】中的程序，如图 1.3 所示。

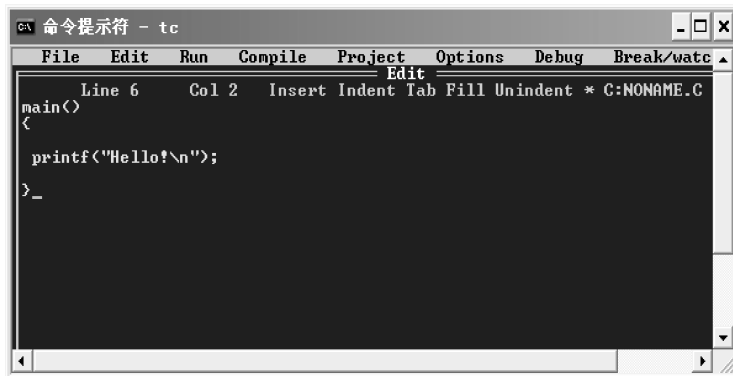


图 1.3 输入【例 1.1】中的程序

(2) 源程序的编辑。在编辑窗口中，可以很方便地对源程序做修改。表 1-2 列出了 Turbo C 的常用编辑命令。

表 1-2 Turbo C 的常用编辑命令

功能键	功 能	功能键	功 能
↑	光标上移一行	Delete	删除光标处的字符
↓	光标下移一行	Backspace	删除光标处的前一个字符
←	光标左移一个字符	Ctrl-Y	删除光标所在行
→	光标右移一个字符	Ins	插入、改写状态转换开关
PgUp	光标上移一页	Ctrl-K-B	定义块首
PgDn	光标下移一页	Ctrl-K-K	定义块尾
Home	光标移到当前行首	Ctrl-K-V	移动定义的块到光标所在处
End	光标移到当前行尾	Ctrl-K-C	复制定义的块到光标所在处
Ctrl-Home	光标移到源程序首	Ctrl-K-Y	删除定义的块
Ctrl-End	光标移到源程序尾	Ctrl-K-H	取消块的定义

3. 保存 C 程序

选择“File/Save”(文件/保存)菜单,或直接按“F2”键,即可保存当前编辑窗口中的 C 程序,保存之前要求先给当前默认文件名为“NONAME.C”的 C 程序重命名。

4. 编译、连接和运行

用任何一种高级语言编写的源程序都不能直接运行。C 源程序在运行之前必须经过编译和连接两个步骤。编译的结果是生成一个扩展名为.OBJ 的目标代码文件,连接的结果则是生成一个扩展名为.EXE 的可执行文件。

选择“Compile”(编译)菜单中的命令可以完成编译和连接操作。更为简单直接的办法则是,选择“Run”(运行)菜单中的“Run”命令,自动完成程序的编译、连接和运行 3 个步骤。

程序运行结束后将自动回到编辑状态。若要看清运行结果,则可按以下步骤进行:

(1) 按“F10”键激活主菜单。

(2) 选择“Run/User Screen”(运行/用户屏幕)菜单,或直接按“Alt-F5”键,即可看到刚才的运行结果,如图 1.4 所示。

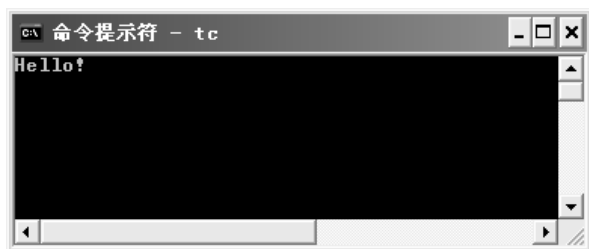


图 1.4 【例 1.1】的运行结果

(3) 按任意键后,可回到编辑状态。

如果编译或连接的过程中出现错误,则出错信息将会显示在“Message”(信息)窗口中。这时,应该根据出错信息的提示,回到编辑状态修改程序。

5. 装入一个已有的 C 源程序

在 Turbo C 中装入一个存储设备上已有的 C 源程序的步骤如下:

(1) 选择“File/Load”(文件/装入)菜单。

(2) 在“Load File Name”提示框中输入要装入的路径名及文件名后按回车键,编辑窗口中即可显示出装入的源程序内容。

6. 结束当前程序,开始编辑另一个 C 程序

结束当前程序编辑另一个程序的步骤如下:

(1) 选择“File/New”(文件/新建)菜单。若当前程序没有保存,则会显示出询问是否保存当前程序的信息,如图 1.5 所示。

(2) 按“Y”键保存程序后,编辑窗口将被清空,这时即可开始建立一个新的 C 源程序。

7. 退出 Turbo C

选择“File/Quit”(文件/退出)菜单,或是直接按“Alt-X”键,即可退出 Turbo C。



图 1.5 询问是否保存当前程序



习题一

1. 填空题

- (1) C 语言程序由_____组成，其中必须有且只能有一个名为_____的函数。C 程序的执行从_____函数开始。
- (2) 每个 C 语句必须以_____号结束。
- (3) 标识符只能由_____、_____和_____三类符号构成，而且标识符的第一个字符必须是_____或_____。
- (4) 关键字是指_____。
- (5) C 程序中，注释的内容应放在_____和_____符号之间。

2. 选择题

- (1) 下面合法的 C 语言标识符是_____。
- A. 3ab B. AB.2 C. a_3 D. #abc
- (2) C 语言中主函数的个数是_____。
- A. 1 个 B. 2 个 C. 3 个 D. 任意个
- (3) 以下有关注释的描述中，错误的是_____。
- A. 注释可以出现在程序中的任何位置。
- B. 程序编译时，不对注释做任何处理。
- C. 程序编译时，要对注释做出处理。
- D. 注释的作用是提示或解释程序的含义，帮助提高程序的可读性。
- (4) C 程序中，main 函数的位置_____。
- A. 必须放在所有函数定义之前。
- B. 必须放在所有函数定义之后。
- C. 必须放在它所调用的函数之前。
- D. 可以任意。

3. 指出并改正下面程序中的错误

(1) main

```
{
    printf("Welcome!");
}
```

(2) main()

```
{
    int a;
    a=5;
    printf("a=%d",a);
}
```

(3) main()

```
{
    int a,b
    a=1,b=2
    printf("%d",a+b)
}
```

4. 分析下列程序，写出运行结果

(1) main()

```
{
    int a,b,c;
    a=2;b=15;
    c=a*b;
    printf("c=%d",c);
}
```

(2) main()

```
{
    printf("Hello!");
    pt();
}
pt()
{
    printf("*****");
}
```

5. 编程题

(1) 编写一个程序，打印出下面的信息：

```
*****
Welcome!
*****
```

(2) 编写一个程序，输入变量 a 和 b 的值，输出表达式 a*b+10 的值。



上机实习指导

一、学习目标

本章是学习 C 语言的入门篇，重点介绍了 C 程序的组成结构和基本的上机操作步骤。通过本章的学习，应建立对 C 语言程序的初步认识，并能在 Turbo C 环境中建立和运行简单的 C 程序。具体要求掌握以下内容：

1. 掌握 C 程序的组成结构和 C 程序的书写格式。
2. 通过上机实习熟悉 Turbo C 环境的操作界面，并初步了解 Turbo C 系统的功能。



3. 掌握基本的上机操作步骤，包括：

- (1) 启动 Turbo C 系统；
- (2) 建立 C 源程序；
- (3) 运行 C 程序；
- (4) 修改 C 程序；
- (5) 保存 C 程序；
- (6) 装入一个已存在的 C 程序；
- (7) 退出 Turbo C 系统。

二、应注意的问题

1. 在程序中使用汉字

编程时有时需要通过程序输入/输出各种中文信息。由于 Turbo C 是在 DOS 环境下运行的，因此，如果需要在 Turbo C 中编辑和运行一个包含了中文信息的 C 程序，就必须在启动 Turbo C 之前运行一个中文 DOS 系统。

目前，使用较普遍的 Windows 操作系统版本是 Windows XP，而在 UC DOS、CC DOS 等多种中文 DOS 系统版本中，与 Windows XP 具有较高兼容性的则是 CCDOS97。键入以下命令即可在运行 CCDOS97 之后启动 Turbo C：

```
C:\>cd ccdos97
C:\CCDOS97>ccdos
C:\CCDOS97>cd\tc
C:\TC>tc
```

2. 同一时刻编辑窗口中只能有一个源程序

当需要建立和运行多个 C 程序时，初学者经常会出现下列错误：一个程序练习结束以后，在该程序后面直接输入另一个程序，这样是达不到目的的。千万注意，同一时刻编辑窗口中只能有一个源程序。一个程序运行结束后，如果想建立并运行另一个程序，则可以选择“File”菜单中的“Load”命令，在弹出的“Load File Name”提示框中输入新的文件名即可，或是选择“File”菜单中的“New”命令，直接开始编辑另一个新程序。

3. 对程序错误的理解

程序错误可以分为两种：编译错误和运行错误。C 语言编译器在编译阶段发现的错误称为编译错误，表现为程序中的语法错误；C 程序通过编译检查后，还需要在实际运行中检查是否有错误，称为运行错误。

编译错误可以通过静态语法检查发现，如声明变量的语法格式不正确等。运行错误不能在静态检查阶段发现，只能在程序运行中才能发现，如 0 做除数等。

一般来说，编译错误出现得更多一些，也更容易排除。运行错误出现得比较少，但比较隐蔽，难以发现。在编写 C 程序的过程中，经常会出现一些错误。对于初学者来说，这些错误似乎是很难逾越的，很难发现。但是只要从一开始就养成良好的编程习惯，严格按照语法要求，这些错误是可以克服的。



上机实习 Turbo C 的基本操作

一、目的要求

1. 掌握 C 程序的基本结构。

2. 熟悉 Turbo C 系统的操作界面。
3. 能熟练地启动 Turbo C 和退出 Turbo C。
4. 掌握在 Turbo C 中建立、运行、修改、保存和装入程序的方法。
5. 掌握插入/删除字符和插入/删除行等基本的编辑操作。

二、上机内容

下面是 3 个从最简单到稍复杂的 C 程序,仔细阅读程序并在 Turbo C 中建立和运行程序,以熟悉 C 程序的基本结构和 Turbo C 的基本操作流程。

1. main()

```
{  
    printf(" Welcome ! ");  
}
```

(1) 在编辑窗口录入该程序后,选择 Run 菜单中的 Run 命令完成程序的编译、连接和运行,最后用 Run 菜单中的 User Screen 命令重现屏幕上的运行结果。

(2) 在函数体中插入一行语句,使上面的程序变成:

```
main()  
{  
    clrscr();    /* 清屏 */  
    printf(" Welcome ! ");  
}
```

再次运行程序并仔细观察运行结果,这次是先清屏,再从屏幕的左上角开始显示“ Welcome !”。

2. main()

```
{  
    int a,b,sum; /* 定义 a、b、sum 3 个变量为 int 类型 */  
    a=1;  
    b=2;  
    sum=a+b; /* 将 a 与 b 之和赋值给 sum */  
    clrscr();  
    printf("sum=%d",sum);  
}
```

(1) 自己先分析程序的运行结果之后再运行该程序,对比自己的判断与屏幕上的结果是否一致,如果有差异,再想想错误出在什么地方。这种做法可以逐步训练自己理解程序和分析程序的能力。

(2) 删除函数体第一行中的变量定义语句“ int a,b,sum;”,或将该语句注释起来,再运行程序,看看会有什么结果。

3. main()

```
{  
    int a,b,c;  
    scanf("%d", &a); /* 输入一个整数到变量 a 中 */  
    scanf("%d",&b); /* 输入一个整数到变量 b 中 */  
    c=max(a,b); /* 调用 max 函数,求 a、b 中的最大值,并把结果赋值给变量 c */  
    printf("%d",c); /* 输出 c 的值 */  
}  
max(a,b) /* 定义 max 函数 */
```



```
int a,b;          /* 定义形式参数 a 和 b */
{
    int t;
    if (a>b) t=a;  /* 求 a、b 中的最大值 */
    else t=b;
    return t;     /* 返回变量 t 的值 */
}
```

程序中的注释是为了帮助对程序的理解，在 Turbo C 中录入程序时可以免去注释部分，以提高建立程序的效率。运行程序后仔细分析运行结果。