

# 第一部分 上机要求与编程环境



本部分介绍了上机实验的目的、要求以及编程环境,以便帮助读者更好地理解并准备进行实验,包含的章节如下。

第 1 章 上机实验的目的和要求

第 2 章 编程环境



## 第 1 章



# 上机实验的目的和要求

本章旨在通过实际操作帮助读者加深对 C 语言程序的理解,提高读者对 C 语言的应用能力和分析问题、解决问题的能力,并培养读者的逻辑思维和算法设计能力。同时,也要求读者掌握 C 语言程序的编写、检查、编译、运行和调试等基本技能。

## 1.1 上机实验的目的

C 语言上机实验的目的主要聚焦于以下几方面,旨在通过实验操作来增强读者的编程技能和理解能力。

(1) 加深理论知识的理解和应用:通过编写和运行 C 语言程序,读者需能够将理论知识(如数据类型、控制结构、函数、指针等)应用到实际编程中,从而加深对这些概念的理解和掌握。

(2) 培养编程能力和解决问题能力:上机实验提供了一个将理论知识转化为实际编程技能的平台。读者需要通过分析问题、设计算法、编写代码、调试程序等一系列过程来解决实际问题,这有助于培养编程能力和解决问题能力。

(3) 熟悉编程环境和工具:在实验过程中,读者需要熟悉并使用 C 语言的编程环境(如 IDE、编译器、调试器等),掌握这些工具的基本操作和使用方法。

(4) 掌握编程规范和风格:通过上机实验,读者可以学习到编写 C 语言程序时应遵循的规范和风格,如代码的可读性、可维护性,注释的添加等。这些良好的编程习惯将对读者在未来的编程生涯产生积极的影响。

(5) 提高调试和查错能力:在编写和运行程序的过程中,读者不可避免地会遇到各种错误。通过上机实验,可以学会使用调试工具来查找和修正错误,从而提高读者的调试和查错能力。这对于读者将来是否能成为一名高效的程序员至关重要。

(6) 增强实践经验和兴趣:通过亲手编写和运行 C 语言程序,读者可以亲身体会编程的乐趣和成就感,进一步激发读者对编程的兴趣和动力。同时,这些实践经验也将为读者未来的学习和工作提供宝贵的参考和借鉴。

## 1.2 上机实验前的准备

C 语言上机实验前的准备工作是确保实验顺利进行的重要环节,主要包括以下几方面。

(1) 准备编程环境:了解计算机上安装的 C 语言的编程环境。掌握编程环境的使用,如编辑、编译、运行和调试 C 语言程序等。

(2) 预习相关知识:了解实验目的、实验步骤和实验要求,掌握与实验内容相关的理论知识、重难点、易错点。确保对即将进行的实验内容有清晰的认识和理论基础。

(3) 熟悉操作界面:提前熟悉编程软件的界面布局、菜单功能、快捷键等,以便在实验过程中提高效率。

(4) 编写程序:①仔细分析实验题目,明确实验要求和目标,根据题目要求,设计算法和数据结构;②根据设计好的算法和数据结构,编写 C 语言源程序,注意代码的规范性、可读性和可维护性;③在编译之前,对源程序进行静态检查,查找并修正能发现的语法错误和逻辑错误。

(5) 准备测试数据:①根据实验要求,准备几组输入数据,这些数据应该尽可能覆盖程序的各种情况,以便全面测试程序的正确性;②对于每组输入数据,都要有明确的预期结果,这有助于在程序运行时验证程序的正确性。

(6) 其他准备:①准备一些相关的参考资料,如 C 语言编程手册、算法图书等,以便在需要时查阅;②准备实验报告的写作材料,包括实验目的、实验步骤、实验结果和实验分析等部分。

## 1.3 上机实验的步骤

上机实验具体包括如下五个步骤。

**步骤 1:** 阅读题目或者程序。对于编程题,应该先看懂题目并分析题意,然后思考用什么方法去解决,整理思路以明确需要用哪些变量保存中间结果以及最后的结果,尽量用伪代码或者流程图等工具写出解题的大概步骤;对于阅读程序题和程序填空题,必须先要认真阅读题目和程序,理解程序并尽量自己试着写出运行结果或者填空。

**步骤 2:** 编辑输入代码。对于编程题,根据前面的分析,编辑输入代码,然后保存代码;对于阅读程序题和程序填空题,把实验题目所给的源代码编辑输入,然后保存代码。

**步骤 3:** 编译程序。把已经输入的程序进行编译,如果出现语法错误,则认真看错误提示,根据提示修改源程序,注意修改之后马上再编译,因为有可能一个地方写错会引起多条编译错误。这里可能需要多次重复“修改→编译”的过程,直到没有编译错误为止(常见编译错误可以参考附录 B)。

**步骤 4:** 运行程序并分析运行结果。运行没有编译错误的程序,然后观察运行结果,注意要对结果进行分析。对于编程题,应该输入多组不同的测试数据来验证编写的程序是否

正确。对于阅读程序题,应该把运行结果与自己的分析结果进行比较,看看是否相同,如果不相同,需要认真地分析是程序输入错了,还是自己分析错了;对于程序填空题,与编程题相似,应该输入多组不同的测试数据来验证编写的程序是否正确。

**步骤 5:** 调试程序,寻找隐藏的错误(bug)。对于编程题,当分析运行结果并发现错误时,需要去检查源程序,找到错误所在。注意,这里语法没有问题,可能是变量引用错误、判断或循环存在逻辑错误,或者算法本身存在错误。出现这种情况时,需要进一步分析题意,检查程序,然后修改程序,再次运行,并分析运行结果。如果此时运行结果还是不对,隐藏的错误一时还找不到,可以借助编译系统的调试工具设置断点,单步调试,查看运行的某些变量的中间结果,进行分析寻找隐藏在程序中的错误。值得注意的是,在上机实验中这一步是最难的,需要有足够的耐心和毅力去克服困难,努力查找隐藏在代码中的错误,实现既定的目标。

## 1.4 实验报告格式

C 语言实验报告是记录学习 C 语言过程中实验活动的重要文档,它不仅反映了实验的过程和结果,还体现了读者分析问题和解决问题的能力,建议提交电子实验报告即可,不用再手写。一个完整的 C 语言实验报告通常应包含以下几个基本部分。

- (1) 解题分析: 包含运用到的知识点、变量及类型定义、主要算法以及解题思路等。
  - (2) 编写代码: 提前或当场在草稿纸上进行编程,建议在预习环节完成。
  - (3) 输入代码: 把编写好的程序代码输入编程软件中,按照程序规范将其编辑并保存为源程序文件。
  - (4) 编译程序: 如果存在错误则需要根据软件提示的错误,对程序进行修改,注意每修改 1 处错误马上重新编译程序,直至编译程序时不再出现错误。
  - (5) 运行程序: 输入测试数据,分析运行结果,当发现结果不对时,可能存在输入输出错误、逻辑错误或者算法错误等。此时需要进一步分析题目和修改程序中隐藏的错误。如果测试时结果都正确,则把运行结果截图保存到实验报告中。
- 实验报告范例如下。

C 语言程序设计实验					
班级		姓名		学号	
实验名称	选择结构程序设计			完成时间	年 月 日
<p><b>1. 实验目的</b></p> <p>(1) 熟练掌握 if 语句的功能和用法,以及 if 语句的嵌套。</p> <p>(2) 掌握 switch 语句的格式、功能和用法。</p> <p>(3) 灵活使用 if 语句和 switch 语句进行选择结构的程序设计。</p>					

## 2. 实验内容

**【题目 1】** 编写一个 C 语言程序,该程序要求用户输入一个整数,然后根据该整数的值,输出该整数是正数、负数,还是零。

(1) 解题分析: 这个问题可以通过使用 if 多分支语句来解决。程序首先接收用户输入的整数,然后使用一个 if-else if-else 结构来判断这个整数与 0 的关系,并根据判断结果输出相应的信息。

(2) 编写代码。

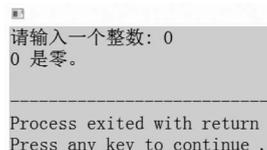
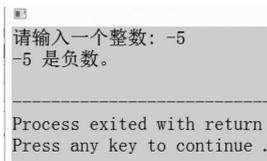
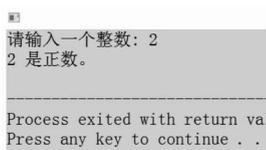
```
#include <stdio.h>
int main()
{
    int number;
    printf("请输入一个整数: ")
    scanf("%d", &number);
    if (number > 0)          // 使用 if 多分支判断并输出结果
        printf("%d 是正数.\n", number);
    else if (number < 0)
        printf("%d 是负数.\n", number);
    else
        printf("%d 是零.\n", number);
    return 0;
}
```

(3) 编译程序。

① 报错信息: 第 6 行报错 expected ';' before 'scanf',意思是 scanf 前缺少分号。

② 解决方法: 在第 5 行代码末尾加分号,表示此行代码结束。

(4) 运行结果截图。



## 【题目 2】

.....

## 【题目 3】

.....

## 3. 实验小结

在本次选择结构程序设计实验中,深入学习了 C 语言中 if 语句的三种分支、switch 语句等选择结构的知识和应用方法,通过一系列的实验操作,不仅加深了对理论知识的理解,还显著提高了解决实际问题的能力。

同时,存在如下问题和需要改进的地方。

(1) 代码优化: 在实验中,部分代码存在冗余和不够简洁的问题,未来需要更加注重代码的优化和重构。

(2) 逻辑复杂性: 在处理一些较为复杂的逻辑判断时,还需要进一步提高自己的思维能力和代码设计能力。

(3) 实际应用: 虽然本次实验涵盖了选择结构的基本应用,但距离实际项目开发中的复杂场景还有一定差距,未来需要多参与实际项目,提升编程实战经验。

## 第 2 章



# 编程环境

## 2.1 Dev-C++ 介绍

Dev-C++是一款专为 Windows 操作系统设计的轻量级 C/C++ 集成开发环境(IDE),它以简洁的界面、强大的功能和易于上手的特点受到广泛欢迎。这款 IDE 集成了代码编辑器、编译器、调试器和项目管理等工具,为开发者提供了一个完整的编程环境。用户可以在 Dev-C++ 中编写、编译、调试和运行 C/C++ 程序,无须在多个软件之间切换,大幅提高了开发效率。

对于需要快速搭建小型项目或进行实验性开发的开发者来说,Dev-C++ 是一个不错的选择。它提供了完整的编程环境和丰富的功能支持,能够满足大多数小型项目和实验性开发的需求。

Dev-C++ 支持 C/C++ 编程语言,并且具有高效的编译和调试功能,这使得它成为算法竞赛和编程练习的理想工具之一。许多算法竞赛选手和编程爱好者都使用 Dev-C++ 进行代码编写和调试。

Dev-C++ 特别适合初学者和学生使用,因为它不仅提供了基本的编程功能,还内置了语法高亮、代码补全等辅助功能,能够帮助用户更快地掌握编程技巧。此外,Dev-C++ 还支持多种语言(包括中文),使得用户可以根据自己的语言习惯进行操作。

## 2.2 Dev-C++ 的安装

Dev-C++ 的安装过程如下。

首先,需要从官方网站下载 Dev-C++ 的安装包。建议下载最新版本的安装文件,以便获得最新的功能和安全性更新。

运行下载好的文件 Setup. exe,等待解压数据,如图 2-1 所示。

解压数据后进入 Installer Language 界面选择语言,如图 2-2 所示。

单击图 2-2 中的 OK 按钮,进入 License Agreement 界面,获取许可协议,如图 2-3 所示。

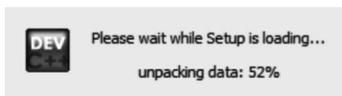


图 2-1 解压数据



图 2-2 Installer Language 界面

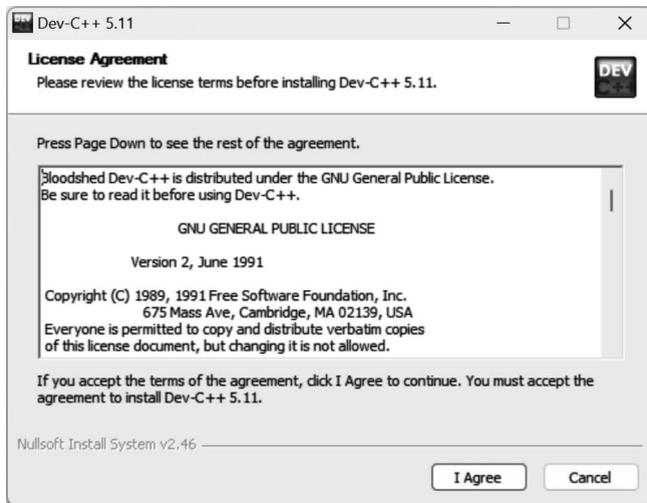


图 2-3 License Agreement 界面

单击图 2-3 中的 I Agree 按钮后进入 Choose Components 界面,选择需要安装的组件,如图 2-4 所示。

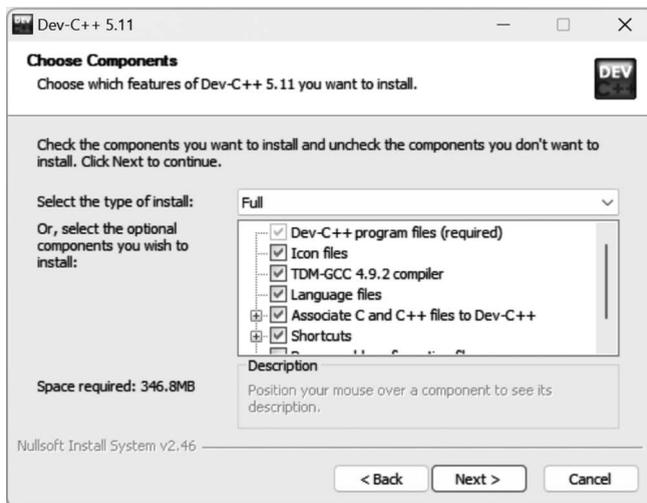


图 2-4 Choose Components 界面

在图 2-4 中单击 Next 按钮后进入 Choose Install Location 界面,如图 2-5 所示。

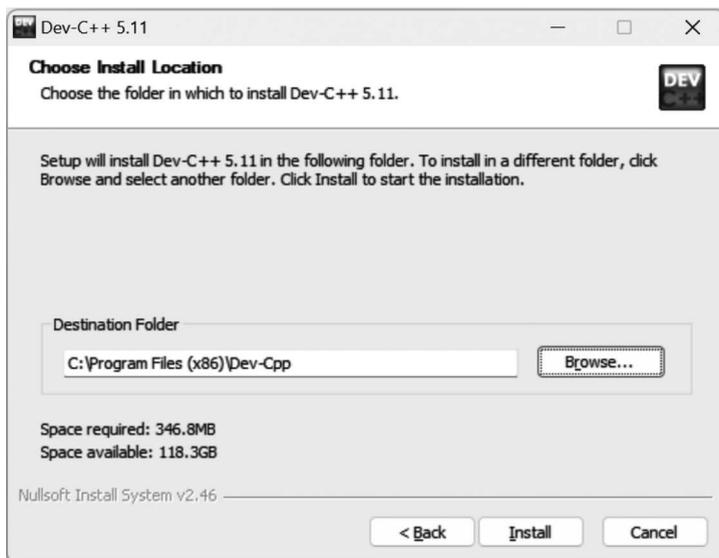


图 2-5 Choose Install Location 界面

在图 2-5 中可选择安装路径,单击 Install 按钮后可自动安装,安装完成后显示安装完成界面,如图 2-6 所示。

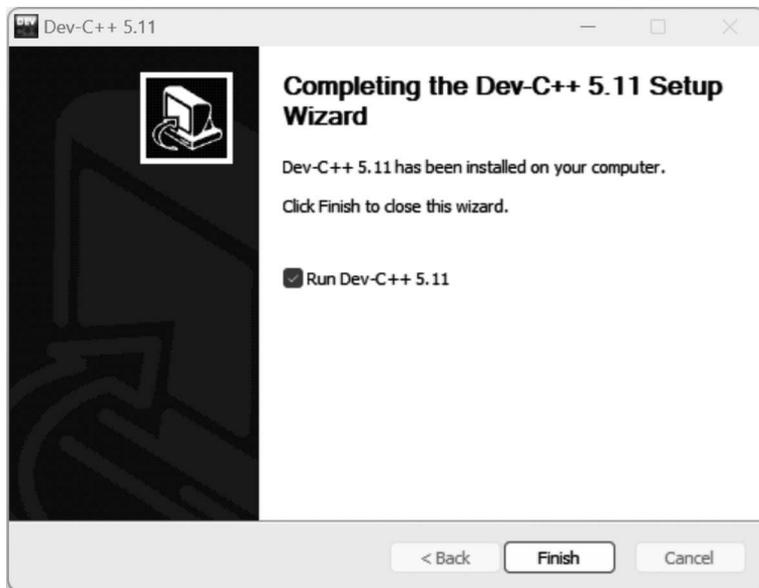


图 2-6 安装完成界面

在图 2-6 中单击 Finish 按钮,Dev-C++ 安装完成并运行 Dev-C++。

## 2.3 代码编辑器的使用

### 2.3.1 新建 C 项目

新建 C 项目的步骤如下。

**步骤 1:** 单击菜单栏上的“文件[F]”→“新建[N]”→“项目[P]”，新建项目如图 2-7 所示。

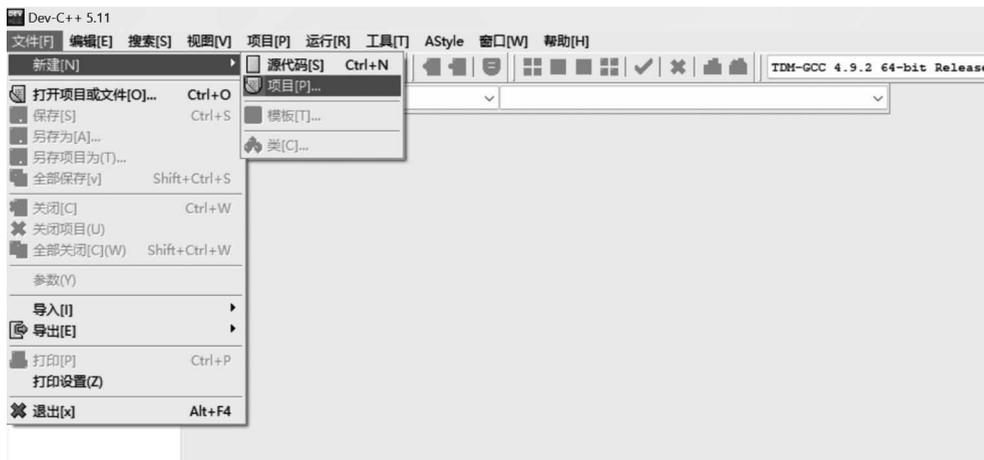


图 2-7 新建项目

**步骤 2:** 选择 Basic 选项卡,选中 Console Application,选中“C 项目”单选按钮,项目名称可以任意命名,完成后单击“确定”按钮,如图 2-8 所示。



图 2-8 新建 C 项目

**步骤 3:** 选择保存项目的路径,文件名可自定义,如图 2-9 所示。

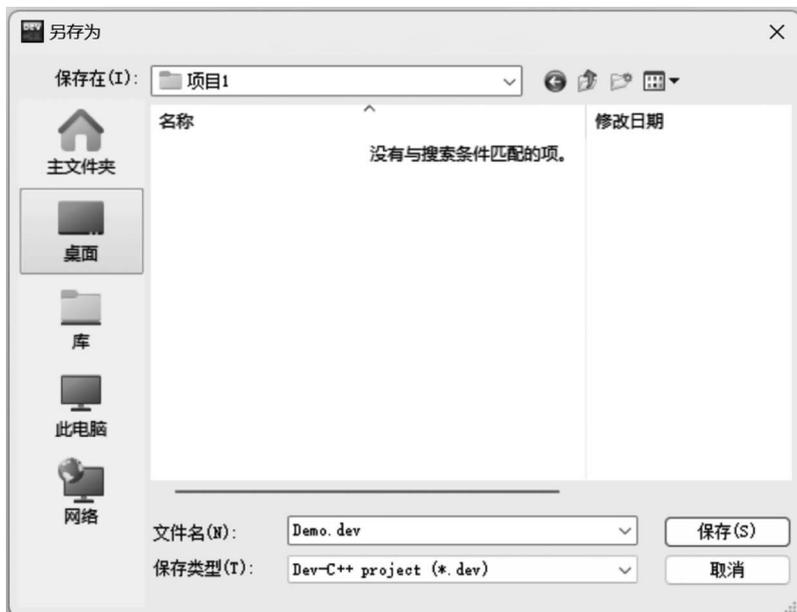


图 2-9 项目保存地址与项目文件名称命名

单击图 2-9 中的“保存”按钮后，C 项目新建完成，显示如图 2-10 所示的 Dev-C++ 主页面。

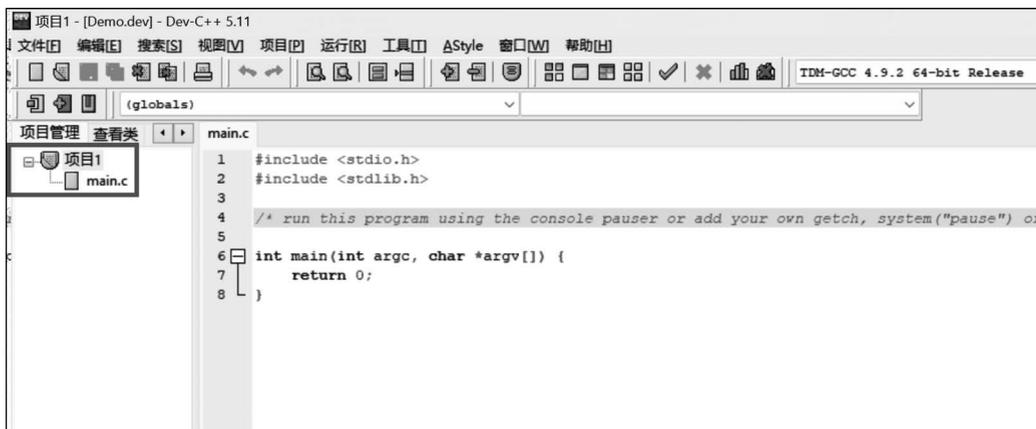


图 2-10 Dev-C++ 主页面

### 2.3.2 新建文件

对于单个源代码的程序，可以直接新建源代码文件。单击图 2-11 中菜单栏上的“文件 [F]”→“新建[N]”→“源代码[S]”，或直接使用快捷键 Ctrl+N，Dev-C++ 主页面中会出现一个新的源文件编辑窗口（“未命名 1”文件），可以在其中编写代码，如图 2-12 所示。

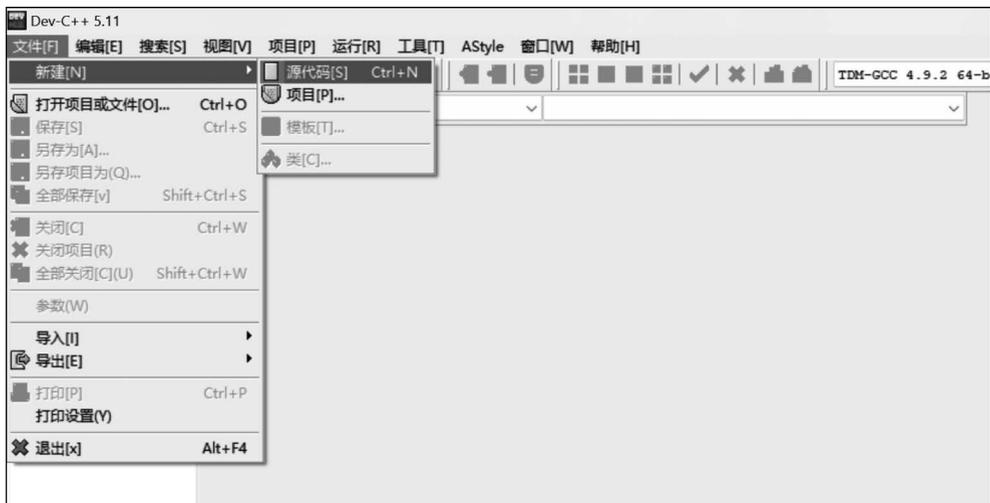


图 2-11 新建源代码

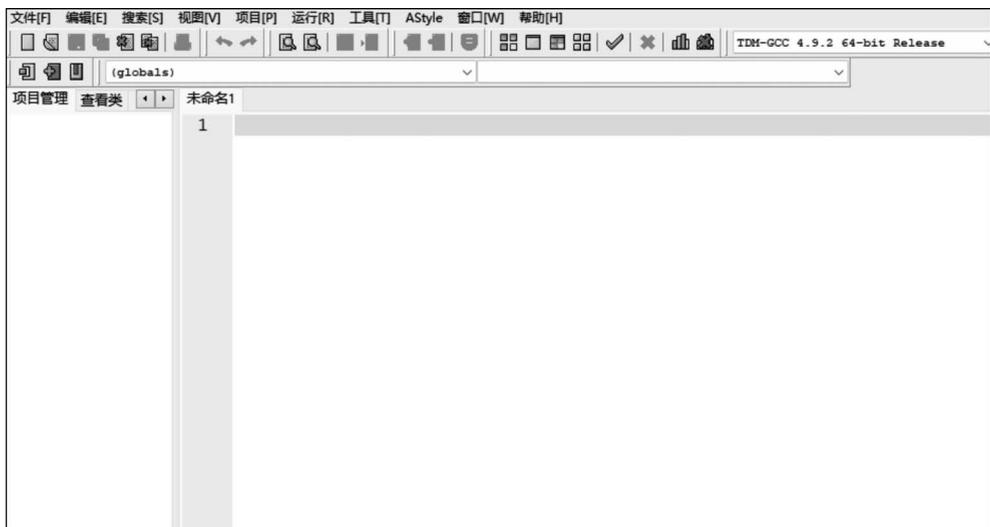


图 2-12 源文件编辑窗口

## 2.4 编译与运行程序

编译与运行程序流程如下。

(1) 编译程序：单击工具栏上的编译(Compile)按钮  或按 F9 快捷键，对程序进行编译，如图 2-13 所示。如果代码没有错误，编译器将生成可执行文件。

(2) 运行程序：编译成功后，单击工具栏上的运行(Run)按钮  或按 F10 快捷键，运行程序，如图 2-14 所示。如果程序包含主函数(main()函数)，则会在控制台窗口显示运行结果。

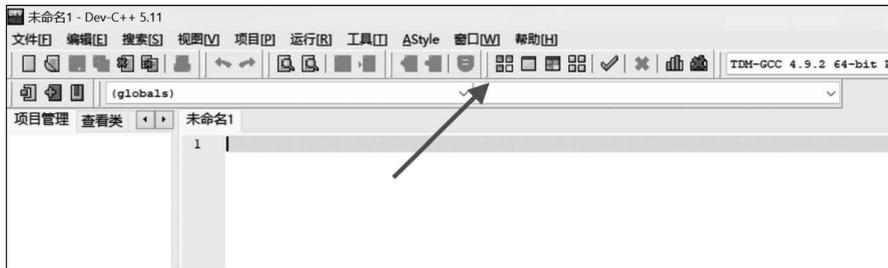


图 2-13 编译程序

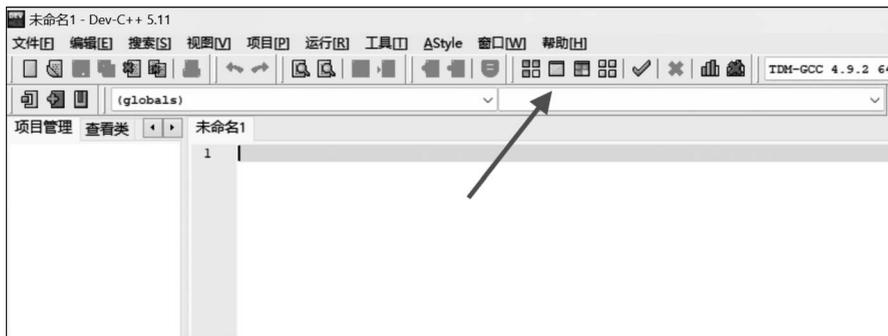


图 2-14 运行程序

## 2.5 调试工具的基本操作

如果程序运行结果不符合预期,可以使用 Dev-C++的调试功能来查找和解决问题。

设置断点:在代码中需要暂停执行的位置单击行号,设置断点。设置断点之后该行为红色底纹显示。

启动调试器:单击工具栏上的调试(Debug)按钮  或按 F5 快捷键,启动调试器。

添加查看:在添加查看里面输入想要查看的变量名。

单步执行:使用下一步(Step)、步入(Step Into)和步过(Step Over)等功能,逐行跟踪代码执行并查看相应变量的值进行分析,寻找里面隐藏的 bug。

## 2.6 其他设置

Dev-C++还提供了许多其他设置选项,如编辑器的字体大小、颜色方案、编译器选项、环境选项等。这些设置可以通过菜单栏的“工具[T]”→“编辑器选项(Editor Options)”或“项目[P]”→“项目选项(Project Options)”等进入设置界面进行调整。还可以通过“工具[T]”→“编译选项(Compiler Options)”设置编译器;通过“工具[T]”→“环境选项(Environment Options)”设置环境选项,例如设置语言等。



# 第二部分 上机实验指导



第二部分包含第 3~20 章,共 18 个实验,实验从简单的程序框架、语法、表达式开始,逐步增加难度,遵循循序渐进的原则,具体章节如下。

- 第 3 章 开发环境的使用
- 第 4 章 数据类型、运算符和表达式的应用(一)
- 第 5 章 数据类型、运算符和表达式的应用(二)
- 第 6 章 数据的输入输出和顺序结构程序设计
- 第 7 章 选择结构程序设计
- 第 8 章 循环结构程序设计(一)
- 第 9 章 循环结构程序设计(二)
- 第 10 章 数组的构造与应用(一)
- 第 11 章 数组的构造与应用(二)
- 第 12 章 函数的应用(一)
- 第 13 章 函数的应用(二)
- 第 14 章 函数的应用(三)
- 第 15 章 函数的应用(四)
- 第 16 章 指针的应用(一)

第 17 章 指针的应用(二)

第 18 章 结构体数据的应用(一)

第 19 章 结构体数据的应用(二)

第 20 章 文件类型的应用



## 第 3 章



# 开发环境的使用

### 【课程思政】

(1) 爱国情怀与科技梦：在探索 C 语言上机实验的初始篇章，不仅需要追溯其作为计算机科学基石的辉煌历程，更需要了解中国在这一领域的迅猛崛起与卓越贡献。每次编写代码，都是对先辈智慧的致敬，也是在播种未来科技的梦想。这不仅是一场技术的学习之旅，更是激发民族自豪感、点燃爱国情怀的壮丽征程。

(2) 自主学习，筑梦编程之旅：在踏入 C 语言世界的起点，不仅要掌握基础的语法与环境的搭建，更要培养自主学习的能力与终身学习的态度。编程之路漫长且充满挑战，每次对未知的探索，都是对自我边界的拓宽。我们应当像探索者一样，勇敢地迈出第一步，主动求知，不懈追求。这不仅是对 C 语言学习的要求，更是对未来职业生涯乃至整个人生的重要准备。自主学习是通往成功的阶梯，而终身学习的态度，则是不断攀登高峰、实现梦想的动力源泉。

## 3.1 实验目的

(1) 了解软件 Dev-C++ 或 VS2020 的基本操作方法，学会独立使用 C 语言程序运行环境。

(2) 熟练掌握在软件 Dev-C++ 或 VS2020 上编辑、编译、连接和运行一个 C 语言程序。

(3) 通过运行简单的 C 语言程序，掌握和理解 C 语言程序的结构，初步了解 C 语言程序的特点。

## 3.2 主要知识点

(1) 如果 C 语言程序需要输入输出功能，一般在程序的最开始加上 `#include <stdio.h>`。

(2) C 语言程序的框架，有且仅有一个主函数，即 `main()` 函数，这是编写 C 语言程序必须要注意的，而且 `main` 不能写错。

(3) `main()` 函数后面必须包含一对花括号,花括号括起来的内容就是函数体,函数体里面是要编写的程序,由若干语句组成,可以实现一定的功能。

(4) 主函数的函数体最后一个语句是“`return 0;`”,这很重要,表示程序正常结束,请在一开始编写程序时就养成写这一个语句的习惯。

### 3.3 实验内容

**【题目 1】** 将如下代码编写进主程序 `test1.c`。

```
#include <stdio.h>      //头文件
int main()              //主函数
{
    printf("欢迎大家,我们一起学习 C 语言!\n");
    return 0;
}
```

编译运行此程序,查看运行结果,并复制运行结果截图至实验报告。

**【题目 2】** 在屏幕上输出如下信息。

```
*****
                我们共同学习,共同进步!
*****
```

(1) 请编写程序实现上述功能。

(2) 编译运行此程序,查看运行结果,并复制运行结果截图至实验报告。

**【题目 3】** 从键盘输入两个整数  $x$  和  $y$ ,输出  $x$  和  $y$  的和。

(1) 请编写程序实现上述功能。

(2) 编译运行此程序,查看运行结果,并复制运行结果截图至实验报告。

**【题目 4】** 从键盘输入两个整数  $x$  和  $y$ ,输出  $x$  和  $y$  的平均值(保留 2 位小数)。

(1) 请编写程序实现上述功能。

(2) 编译运行此程序,查看运行结果,并复制运行结果截图至实验报告。

### 3.4 思考与练习

(1) 在编写程序之后,应该先保存程序,保存时注意选择保存的位置,如何保存 C 语言程序?

(2) 在编写并保存好一个程序之后,编译程序,当发现程序中有错误,如何最快找到错误并修改它?

(3) 在编写并保存好一个程序之后,一般先编译程序,程序没有错误再运行程序,也可以编译和运行一起完成。请问在 Dev-C++ 中,除了可以通过菜单实现,还可以按什么快捷键编译程序、运行程序,以及编译运行程序?

(4) 试着编写一个程序,从键盘输入两个整数,然后求这两个整数的最大值并将其输出。



### 【课程思政】

(1) 编织代码梦,责任照亮未来:在变量的编写中,每次精准的命名与细心的初始化,都要如同织锦般细腻,赋予代码以生命和灵魂。这不仅是技术的展现,更是责任心的体现。每行代码都是对未来的承诺,它们将影响程序的稳定性与可维护性。因此,编程培养的是一种责任感,一种对未来负责的态度,让代码成为心中最精致的艺术品。

(2) 绿色编程,共绘地球蓝图:数据类型的选择,不仅是编程技巧的体现,更是对资源高效利用的深刻理解。倡导绿色编程,旨在通过优化代码,减少内存浪费,为地球减负。这不仅是技术的追求,更是对可持续发展的贡献。在每个编程的细微之处,种下绿色的种子,期待它们在未来长成参天大树,为地球家园增添一片绿意。

## 4.1 实验目的

- (1) 熟悉 C 语言数据类型,熟练掌握整型、字符型和实型变量的定义。
- (2) 掌握变量的使用方法,掌握变量赋值的基本方法。
- (3) 进一步熟悉 C 语言程序的集成编译环境,积累上机实验的经验。

## 4.2 主要知识点

(1) 整型常量的表示方法:十进制、八进制和十六进制方式。十进制最常见,八进制以 0 开头,十六进制以 0X 开头。大家不仅要掌握整型常量常见的三种表示方法,还要掌握它们之间的相互转换。注意,C11 标准支持二进制表示的整型常量,由 0b 或者 0B 开头。

(2) 变量有三要素:变量类型,变量名,变量值。变量类型决定了变量可以存储怎样的数据以及数据的范围,而变量名是访问该变量的唯一标识,变量值是变量里面存储的具体值。变量的定义方法:

类型名 变量名;

赋值方法: