

巍巍交大 百年书香
www.jiaodapress.com.cn
bookinfo@sjtu.edu.cn



策划编辑 杨 洋
责任编辑 胡思佳
封面设计 黄燕美



C语言 程序设计项目教程

C YUYAN CHENGXU SHEJI XIANGMU JIAOCHENG



高等职业教育计算机系列创新教材

C语言
程序设计项目教程

主编 蒋高飞 张凌雪



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS



高等职业教育计算机系列创新教材

C语言 程序设计项目教程

主编 蒋高飞 张凌雪



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

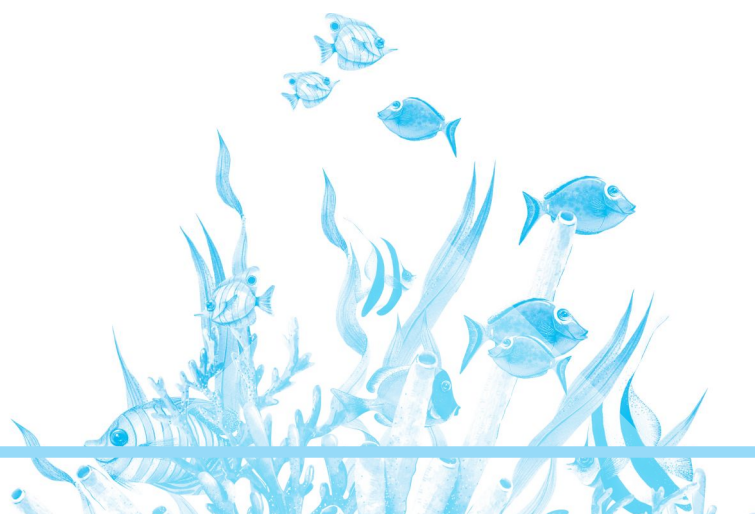


扫描二维码
关注上海交通大学出版社
官方微信

ISBN 978-7-313-30511-4



定价: 55.00元

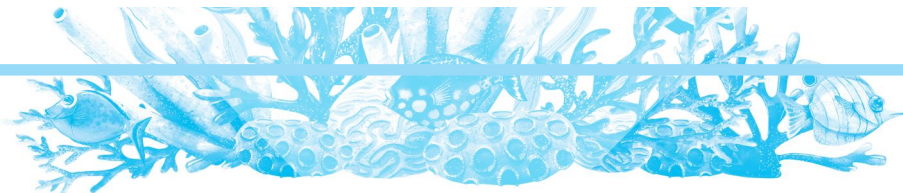


高等职业教育计算机系列创新教材

C语言

程序设计项目教程

主 编 蒋高飞 张凌雪
副主编 杨天时 张 楠



上海交通大学出版社
SHANGHAI JIAO TONG UNIVERSITY PRESS

内容提要

本书共 10 个项目,分别为“显示小鸡吃虫子动画背景”的设计与实现、“显示动画中的小鸡”的设计与实现、“小鸡向右走”动画的设计与实现、“小鸡吃虫子”动画的设计与实现、“显示多条虫子”动画的设计与实现、“小鸡吃虫子”动画函数形式的设计与实现、“小鸡吃虫子”动画指针形式的设计与实现、“多条虫子”动画的结构体形式的设计与实现、“小鸡吃虫子存储分数”的设计与实现和“小鸡吃虫子”综合项目的实现。

本书可作为高等职业教育 C 语言程序设计课程的教材使用,也可供相关人员参考使用。

图书在版编目(CIP)数据

C 语言程序设计项目教程 / 蒋高飞, 张凌雪主编. —
上海: 上海交通大学出版社, 2024. 8
ISBN 978-7-313-30511-4

I. ①C… II. ①蒋… ②张… III. ①C 语言—程序设计—
计—高等职业教育—教材 IV. ①TP312.8

中国国家版本馆 CIP 数据核字(2024)第 062769 号 9 号

C 语言程序设计项目教程
C YUYAN CHENGXU SHEJI XIANGMU JIAOCHENG

主 编: 蒋高飞 张凌雪

出版发行: 上海交通大学出版社

邮政编码: 200030

印 制: 三河市骏杰印刷有限公司

开 本: 850 mm×1 168 mm 1 / 16

字 数: 417 字

版 次: 2024 年 8 月第 1 版

书 号: ISBN 978-7-313-30511-4

定 价: 55.00 元

地 址: 上海市番禺路 951 号

电 话: 021-64071208

经 销: 全国新华书店

印 张: 17.25

印 次: 2024 年 8 月第 1 次印刷

版权所有 侵权必究

告读者: 如您发现本书有印装质量问题请与印刷厂质量科联系

联系电话: 0316-3662258

Preface

前言

本书以高等职业教育的职业化，重视学生实际动手操作能力为出发点，以动画项目为主线、以案例为依托、以学生兴趣为导向，内容讲解循序渐进、由浅入深，适合高等职业教育学生的学习需求。

本书的编写特色如下。

1. 兴趣导向的动画项目为教材主线

为了提高学生的学习兴趣，本书以“小鸡吃虫子”动画为主线，根据动画实现步骤划分成九个项目，每个项目都能实现一个完整功能；后一个项目在前一个项目功能的基础上叠加新功能，在项目 10 中把前面所有的项目功能整合到一起，组成一个完整功能的程序。

在每个项目中讲解相关理论知识点，在动画功能叠加的基础上，实现知识点的叠加，二者相辅相成、依次递进。这样设计的教程结构既让学生在兴趣中潜移默化地掌握了理论知识，又从需求分析到功能实现，引导学生设计一个完整的 C 语言程序，达到事半功倍的学习效果。

2. 理实一体的授课过程融入教材

本书每个项目均采用 [项目要求] → [项目描述] → [项目准备] → [相关知识] → [项目实施] → [项目延伸] → [评价反馈] → [巩固练习] 的编写结构，在 [相关知识] 中讲解的每一个知识点又细化为 [引导案例] → [知识点解析] → [练一练] → [做一做] 四个环节，在书中把理实一体的课堂教学过程完整体现出来，让学生在实践中掌握理论知识。

3. 强调实践能力和创新思维能力

在主要例题讲解完后，设置了进一步思考的拓展问题，让学生养成爱思考的习惯，能带着开拓性的新思维去解决问题，并能提出更高层次的问题。

在 [做一做] 和 [项目实施] 中让学生根据提示完成程序的编写、调试等环节，以提高学生的分析问题能力、解决问题能力和动手操作能力。

4. 课证结合

在 [巩固练习] 中以历年计算机等级考试（二级 C 语言）的部分真题作为课后习题，帮助学生了解等级考试的相关考题信息。

5. 融合思政

教书的同时要育人，为党育人、为国育才。本书结合党的二十大精神在每个项目中添加了思政要求，让学生在学习知识的同时提高自身素质，实现德智体美劳全面发展。

随书提供与教材相关电子教学资源，如教学课件、例题程序、项目程序和习题答案等；书中



所有程序都已在“Dev-C++”中调试且可以正确运行，读者可以放心引用本书中的代码。

本书由吉林交通职业技术学院的蒋高飞、张凌雪老师担任主编，吉林交通职业技术学院的杨天时、张楠老师担任副主编，长春职业技术学院的杨柏婷老师和中兴新思教育的张帅工程师参编，由蒋高飞老师统稿。

由于编者水平有限，书中难免存在疏漏之处，恳请读者批评指正。

编 者

Contents

目录



项目 1

“显示小鸡吃虫子动画背景”的设计与实现

1

项目要求	1
项目描述	2
项目准备	3
相关知识	4
1.1 了解 C 语言	4
1.2 图形文件	11
项目实施	17
项目延伸	18
评价反馈	19
巩固练习	19



项目 2

“显示动画中的小鸡”的设计与实现

21

项目要求	21
项目描述	22
项目准备	23
相关知识	24
2.1 格式输入输出	24
2.2 字符输入输出	34
项目实施	39
项目延伸	40
评价反馈	40
巩固练习	41



项目 3

“小鸡向右走”动画的设计与实现

43

项目要求	43
------------	----



项目描述	44
项目准备	45
相关知识	46
3.1 if 选择语句	46
3.2 switch 选择语句	53
项目实施	58
项目延伸	59
评价反馈	60
巩固练习	61



项目 4

“小鸡吃虫子”动画的设计与实现 65

项目要求	65
项目描述	66
项目准备	67
相关知识	68
4.1 单循环语句	68
4.2 中断语句	75
4.3 循环嵌套	79
项目实施	84
项目延伸	86
评价反馈	86
巩固练习	87



项目 5

“显示多条虫子”动画的设计与实现 91

项目要求	91
项目描述	92
项目准备	93
相关知识	94
5.1 一维数组	94
5.2 二维数组	98
5.3 字符串	105
项目实施	113
项目延伸	115

评价反馈	115
巩固练习	116



项目 6

“小鸡吃虫子”动画函数形式的设计与实现 119

项目要求	119
项目描述	120
项目准备	121
相关知识	122
6.1 函数的定义和调用	122
6.2 数值传递	127
6.3 地址传递	134
6.4 变量作用域和存储类别	139
6.5 递归调用	145
项目实施	151
项目延伸	153
评价反馈	153
巩固练习	154



项目 7

“小鸡吃虫子”动画指针形式的设计与实现 159

项目要求	159
项目描述	160
项目准备	161
相关知识	162
7.1 指向普通变量的指针	162
7.2 指向一维数组的指针	166
7.3 指向字符串的指针	170
7.4 指针指向二维数组	175
7.5 指针做函数参数	183
项目实施	190
项目延伸	191
评价反馈	192
巩固练习	192





项目 8

“多条虫子”动画结构体形式的设计与实现

197

项目要求	197
项目描述	198
项目准备	199
相关知识	200
8.1 结构体变量	200
8.2 结构体数组	207
8.3 结构体指针	214
项目实施	220
项目延伸	222
评价反馈	223
巩固练习	223



项目 9

“小鸡吃虫子存储分数”的设计与实现

227

项目要求	227
项目描述	228
项目准备	229
相关知识	230
9.1 文件操作	230
9.2 文件定位	242
项目实施	251
项目延伸	252
评价反馈	253
巩固练习	253



项目 10

“小鸡吃虫子”综合项目的实现

257

项目要求	257
项目描述	258
项目实施	258
项目程序	258
项目延伸	262

附录

263

参考文献

268

项目 1

“显示小鸡吃虫子动画背景”的设计与实现

项目要求

知识要求

- ◎ 掌握 C 语言的基本语法规则。
- ◎ 掌握 C 语言程序的基本结构。
- ◎ 掌握 C 语言的图形坐标系和基本图形文件。

技能要求

- ◎ 熟悉 C 语言编辑器 Dev-C++ 的基本使用方法。
- ◎ 能用 C 语言编辑器编辑、调试和执行简单的 C 语言程序。

思政要求

- ◎ 具有谦虚谨慎的精神。
- ◎ 具有自信自强的精神。



项目描述

本项目实现“小鸡吃虫子”动画的第一步——显示动画背景，如图 1-1 所示。



图 1-1 显示动画背景

实现方法：

1. 设置图形界面

利用初始化图形界面函数 `initgraph()` 进入 C 语言的图形模式，并设置图形界面的显示尺寸。

2. 显示背景图片

定义装载图片的图形变量，利用载入图像函数 `getimage()` 装载背景图片，再利用输出图像函数 `putimage()` 在图形界面的指定坐标位置显示背景图片。

实现以上功能都涉及 C 语言中图形函数的使用，但是只使用独立函数并不能实现所需的功能，需要把各种函数和语句按照 C 语言的语法规则编写成合格的程序，才能在 C 语言编辑器中正确运行，得出预定的执行结果。

因此在本项目中学习：

- (1) C 语言语句的书写格式、规则；
- (2) C 语言程序的基本组成；
- (3) 创建 C 语言的图形坐标系；
- (4) C 语言中定义图片变量、载入图片函数及显示图片函数等。

通过学习以上知识了解 C 语言的简单控制台程序和图形程序的编写流程和基本算法。



项目准备

根据【项目描述】中讲述的本项目中需要掌握的知识点，通过个人学习、小组讨论等方式完成以下预习任务：

(1) 写出 C 语言程序的基本结构。

(2) 画出 C 语言图形界面的坐标系。

(3) 写出 C 语言图形界面载入图片和显示图片的函数，并举例。





相关知识



1.1 了解 C 语言

1.1.1 引导案例

在 Dev-C++ 编辑器中显示“Hello C 语言!”。

【案例分析】

(1) 在 Dev-C++ 编辑器（下载地址和安装见 1.1.2 知识点解析）中编辑、调试、执行程序并显示结果，Dev-C++ 编辑器的执行界面及各区域功能如图 1-2 所示。



图 1-2 Dev-C++ 编辑器界面

(2) 用 C 语言的输出函数 `printf()` 实现在显示器上显示输出信息，该函数是系统函数，包含在头文件 `stdio.h` 中。C 语言语法规定要在程序的第一行添加头文件，才能使用该头文件中包含的系统函数。

【操作步骤】

(1) 打开 Dev-C++ 编辑器，执行“文件”→“新建”→“源代码”命令，创建一个新的 C 语言程序源文件，如图 1-3 所示；或单击工具栏中的“新建”按钮；或按快捷键 `Ctrl+N` 也可以创建 C 语言程序的源文件。

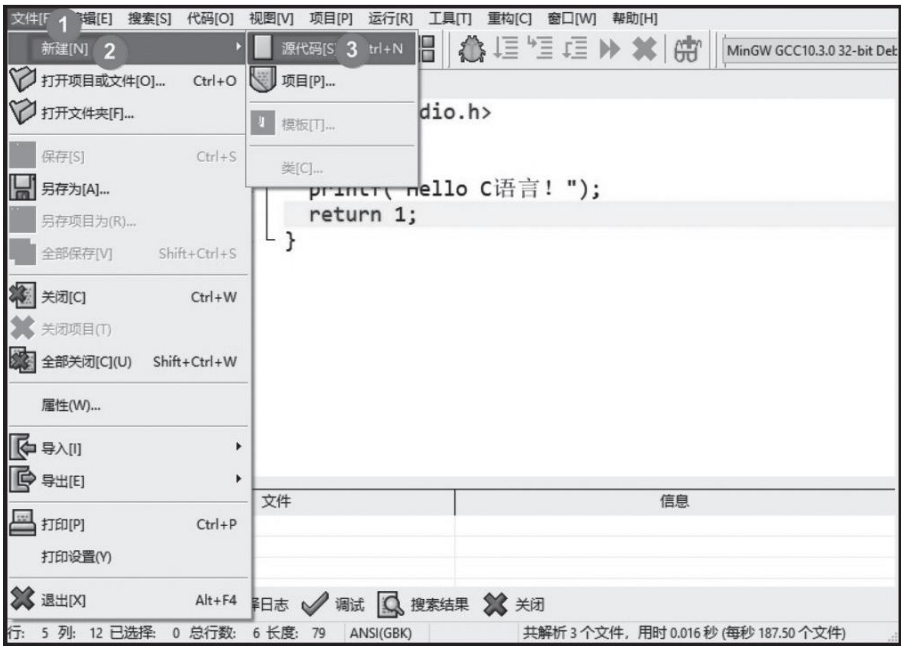


图 1-3 创建 C 语言程序文件

(2) 在代码编辑区输入代码 1-1。

代码 1-1

行 号	代 码	功 能
1	#include<stdio.h>	头文件，包含 printf() 输出函数
2	int main()	主函数首部，main 是主函数名字，固定不变
3	{	主函数体开始
4	printf("Hello C 语言! ");	屏幕上显示信息 "Hello C 语言! "
5	return 0;	返回值 0，程序正常结束
6	}	主函数体结束

(3) 执行“运行”→“运行”/“编译运行”命令，如图 1-4 所示，或者单击工具栏中的“编译运行”按钮运行代码，若程序有错误，修改后再次执行“编译运行”命令，直至能正确运行并得出正确结果。程序正确执行后的结果如图 1-5 所示。

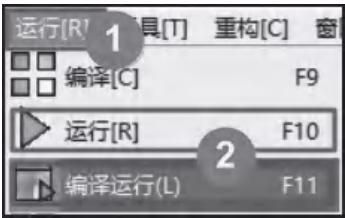


图 1-4 执行程序步骤

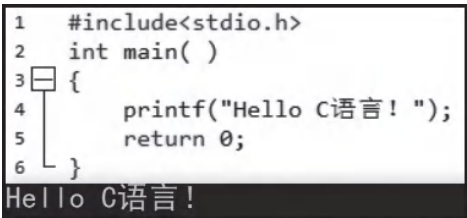


图 1-5 程序执行结果



1.1.2 知识点解析

1. 编程语言的概念

编程语言是符合语法规则的固定词汇组成的一系列命令，通过这些命令向计算机（或计算设备）发出指令，与计算机沟通来实现指定的任务。编程语言一般分为机器语言、汇编语言和高级语言。

（1）机器语言。因为计算机内部只能处理二进制代码，用二进制代码 0 和 1 描述的指令称为机器指令，全部机器指令的集合构成计算机的机器语言。用机器语言编写的程序为目标程序，只有目标程序才能被计算机直接识别和执行。

但是机器语言编写的程序难以记忆，不便于阅读和书写，且依赖于具体编程环境，局限性很大，因此把机器语言归为低级语言。

（2）汇编语言。汇编语言和机器语言一样，也是直接对硬件操作。汇编语言源程序生成的可执行文件比较小且执行速度很快，指令采用了英文缩写的标识符，相对于机器语言更容易识别和记忆。

但是汇编程序的每一句指令只能对应实际操作过程中的一个很细微的动作，因此汇编程序比较冗长、复杂且容易出错。

（3）高级语言。高级语言是一种独立于机器，面向过程或对象的语言。高级语言编写的源程序相对低级语言有较高的可读性，更易理解；但是需要经过编译或解释成机器语言（目标程序），计算机才能执行。

高级语言包括很多种，如 Java，C++，C#，Python 等，每种编程语言都有自己的语法规则。

2. 了解 C 语言

1963 年英国剑桥大学推出 CPL（combined programming language）。

1967 年英国剑桥大学的马丁·理查兹（Martin Richards）对 CPL 语言进行简化，产生了 BCPL（basic combined programming language）。

1970 年美国贝尔实验室的肯·汤普森（Ken Thompson）进一步简化 CPL，提出了 B 语言。

1972 年美国贝尔实验室的丹尼斯·里奇（Dennis Ritchie）对其完善和扩充，提出了 C 语言。

1989 年美国国家标准研究所（ANSI）制定了第一个 C 语言标准“ANSI C”，也称 C89 标准。

2000 年国际标准化组织（ISO）和国际电工委员会（IEC）采纳了第二个 C 语言标准，简称 C99 标准。

2011 年国际标准化组织和国际电工委员会采纳了第三个 C 语言标准，简称 C11 标准。

C 语言的诞生与 UNIX 操作系统的开发是密不可分的。UNIX 最初是肯·汤普森用汇编语言编写的，在丹尼斯·里奇提出 C 语言之后，他们将原来的 UNIX 用 C 语言改写，使其修改、移植更便利。随着 UNIX 操作系统的广泛应用，C 语言也跟着流行起来。在 Windows 操作系统出现后，在 C 语言的基础上扩展了许多界面可视化设计的面向对象编程语言，如 Visual C++，C# 等。

C 语言是结构化、模块化的编程语言，语句简洁、紧凑，语法限制不太严格，程序设计自由度高，使用方便、灵活；运算符和数据结构丰富，表达式能力强，便于数据的描述与存储；允许直接访问物理地址，对硬件进行操作；生成的目标代码质量高，程序执行效率高，可移植性好。



因此 C 语言在操作系统和驱动程序的系统软件开发、嵌入式和单片机应用开发及游戏引擎开发等领域的应用非常广泛。2024 年 2 月 TIOBE 公布的编程语言排行榜中 C 语言排在第 2 位，如图 1-6 所示。

Feb 2024	Feb 2023	Change	Programming Language	Ratings	Change
1	1		Python	15.16%	-0.32%
2	2		C	10.97%	-4.41%
3	3		C++	10.53%	-3.40%
4	4		Java	8.88%	-4.33%
5	5		C#	7.53%	+1.15%
6	7	^	JavaScript	3.17%	+0.64%
7	8	^	SQL	1.82%	-0.30%
8	11	^	Go	1.73%	+0.61%
9	6	v	Visual Basic	1.52%	-2.62%
10	10		PHP	1.51%	+0.21%

图 1-6 C 语言在编程语言中的排名

3.C 语言程序的基本结构

一个程序就是一个算法的具体实现。算法（algorithm）是指解题方案的准确而完整的描述，是一个有穷的规则集合。算法有五个重要特性：有穷性、确切性、可行性、输入项（有 0 个或多个输入）、输出项（有一个或多个输出）。

一个 C 语言程序是由一个或多个函数组成的，其中必须包含一个 main 函数（有且仅有一个 main 函数），每个函数在程序中的位置可以是任意的，但一定是从主函数开始执行，并在主函数结束整个程序的执行。

一个函数包括函数首部 and 函数体两个部分，可参考代码 1-2 所示。

（1）函数首部：函数第一行，包含函数返回值类型、函数名、函数参数（称为形式参数）。

（2）函数体：函数首部下面花括号内的部分，包含声明语句、输入或赋值语句、执行语句和输出语句。

C 语言中的语句用分号作为语句结束标志；用符号“//”（行注释）或者“/*...*/”（段注释）对语句作注释，注释语句不执行。



C语言程序的基本结构-r30e3l

代码 1-2

行 号	代 码	功 能	程序结构
1	#include"stdio.h"	// 头文件	预处理命令
2	int main()	// 主函数首部	函数首部
3	{	// 主函数体开始	函数体
4	int a,b,max;	// 声明变量 a、b、max	声明变量
5	printf(" 输入两个数: ");	// 输入提示	输入
6	scanf("%d%d",&a,&b);	// 给变量 a 和 b 输入值	





续表

行 号	代 码	功 能	程序结构
7	if(a>b)	// 判断 a 和 b 的值,若 a>b	执行
8	max=a;	// 把 a 值赋给 max	
9	else	// 若 a<b	
10	max=b;	// 把 b 值赋给 max	
11	printf(" 两数中大数 =%d",max);	// 显示两数中最大值 max	输出
12	return 0;	// 返回值 0, 程序正常结束	
13	}	// 主函数体结束	

4. 安装 Dev-C++ 编辑器

Dev-C++ 是免费的 C/C++ 编辑器，界面和操作都非常简单，可在网址 <https://sourceforge.net/projects/orwelldvcpp/> 下载，如图 1-7 所示。



图 1-7 编辑器 Dev-C++ 下载界面

以 Dev-C++ 5.11 为例讲述安装步骤。

- (1) 双击安装包中的“Dev-Cpp Setup.exe”文件，进入启动安装界面，如图 1-8 所示。
- (2) 安装完后在图 1-9 所示界面中先选择“English”，单击“OK”按钮，后续安装步骤再改成中文。

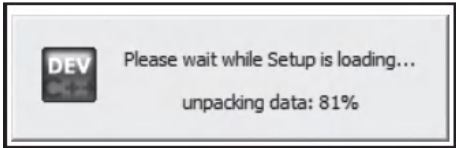


图 1-8 Dev-C++ 启动安装界面

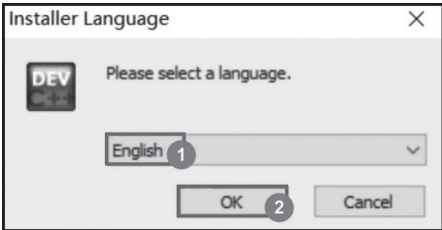


图 1-9 选择语言

- (3) 在图 1-10 所示界面单击“I Agree”按钮，进入下一步；在图 1-11 所示界面单击“Next”按钮，进入下一步。

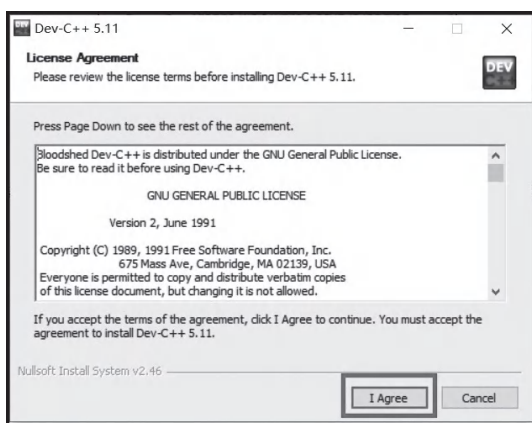


图 1-10 单击“I Agree”按钮

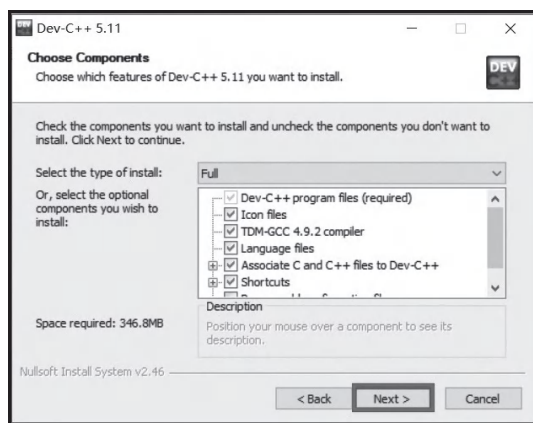


图 1-11 单击“Next”按钮

(4) 在图 1-12 所示界面选择安装路径, 单击“Install”按钮, 开始安装 Dev-C++。

(5) 在图 1-13 所示界面中单击“Finish”按钮结束安装, 开始运行 Dev-C++。

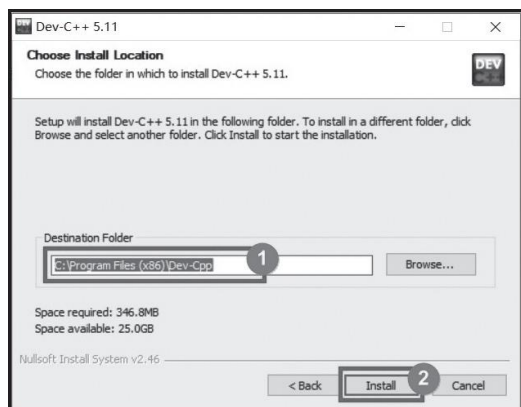


图 1-12 开始安装界面图



图 1-13 安装结束界面

(6) 在图 1-14 所示界面中选择操作界面语言“简体中文/Chinese”, 单击“Next”按钮, 进入 Dev-C++ 的程序编辑界面, 如图 1-15 所示。

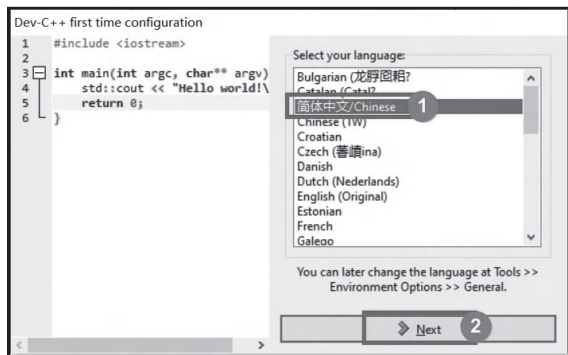


图 1-14 选择编辑器语言

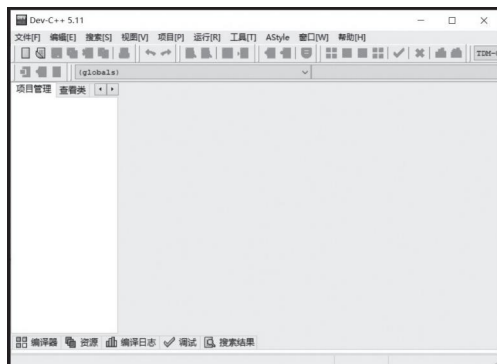


图 1-15 Dev-C++ 初始界面

1.1.3 做一做

屏幕上显示由符号“*”组成的“菱形”图案, 如图 1-16 所示。



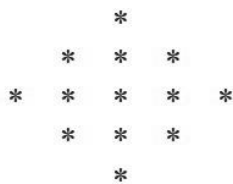


图 1-16 “菱形” 图案

【算法引导】

用 printf() 输出函数显示多行 “*” 即可。如第一行输出一个 “*” 的语句为 printf(" *\n"); (“\n” 是回车换行)。

【算法流程】

行 号	算法流程
1	头文件
2	主函数首部
3	{
4	printf() 函数输出 1 个 “*”
5	printf() 函数输出 3 个 “*”
6	printf() 函数输出 5 个 “*”
7	printf() 函数输出 3 个 “*”
8	printf() 函数输出 1 个 “*”
9	返回值 0
10	}

根据上述算法流程给出该程序中关键的第 4 行至第 6 行参考代码 1-3。

代码 1-3

行 号	代 码
4	printf(" *\n");
5	printf(" ***\n");
6	printf(" *****\n");

【算法实现】

参考 “算法引导” “算法流程” 及代码 1-3，自主编写、调试、执行代码，锻炼学习思维和动手操作能力，执行结果正确后把正确代码写到下方。



1.2 图形文件

1.2.1 引导案例

在 Dev-C++ 编辑器显示一张图片。

【案例分析】

在 C 语言编辑器中显示图片需要进入图形模式，用 `putimage()` 函数输出图片，该函数在头文件 `graphics.h` 中。

【操作步骤】

(1) 打开 Dev-C++ 编辑器，执行“文件”→“新建”→“源代码”命令，创建一个新的 C 语言程序源文件；或单击工具栏中的“新建”按钮；或按快捷键 `Ctrl+N` 也可以创建程序源文件。

(2) 在代码编辑区输入参考代码 1-4。

代码 1-4

行 号	代 码	功 能
1	<code>#include "graphics.h"</code>	头文件
2	<code>int main()</code>	主函数首部
3	<code>{</code>	主函数体开始
4	<code>initgraph(600,300);</code>	设置图形界面宽为 600，高为 300
5	<code>PIMAGE fj;</code>	定义存储图片变量 fj
6	<code>int fj_x=0,fj_y=0;</code>	定义显示图片的 x 和 y 坐标
7	<code>fj=newimage();</code>	初始化图片变量 fj
8	<code>getimage(fj,"fengjing.jpg");</code>	给图片变量 fj 装载图片
9	<code>putimage(fj_x, fj_y,fj);</code>	在坐标 (fj_x,fj_y) 位置显示图片
10	<code>getch();</code>	暂停程序执行
11	<code>return 0;</code>	返回 0
12	<code>}</code>	主函数体结束

注意

第 8 行代码 `getimage(fj,"fengjing.jpg");` 中引号里只有加载的图片文件名，此时 C 语言程序（即扩展名为 `.cpp` 的源文件）和图片在同一个文件夹中。

若图片单独在另一个文件夹（如“image”）中，此时需要在引号里的图片名称前加上图片所在文件夹的路径，语句为 `getimage(fj,"image/fengjing.jpg");`。



(3) 单击工具栏中的“编译运行”按钮运行代码,若有错误,修改后再次运行,直至代码能正确运行。程序执行结果如图 1-17 所示。



图 1-17 程序执行结果

1.2.2 知识点解析

1. 了解 EGE

EGE (easy graphics engine) 是 Windows 下的简易绘图库,支持 VC 系列、C-Free、Dev-C++ 等 IDE。可在官网 <https://xege.org> 下载,如图 1-18 所示。



图 1-18 EGE 下载界面

将 EGE 安装到 Dev-C++ (若安装目录为“C:\Dev-Cpp”)中的具体步骤如下。

(1) 将 EGE 解压缩后的 include 文件夹中的“ege.h”和“graphics.h”两个文件复制到“C:\Dev-Cpp\MinGW64\x86_64-mingw32\include”目录下。

(2) 将目录“lib\mingw64\lib”里面的“libgraphics64.a”文件复制到“C:\Dev-Cpp\MinGW64\x86_64-mingw32\lib”目录下。

(3) 打开 Dev-C++, 执行“工具”→“编译选项”命令,如图 1-19 所示,在打开的“编译器选项”对话框中切换到“编译器”选项卡,在“在链接时加入以下参数”中填入字符串“-lgraphics64 -lgdiplus -luuid -lmsimg32 -lgdi32 -lmm32 -lole32 -loleaut32 -lwmm”,如图 1-20 所示。

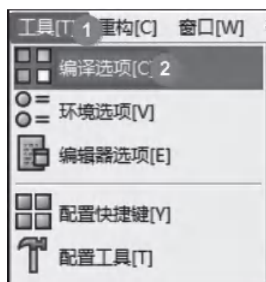


图 1-19 “编译选项”子菜单



图 1-20 设置 EGE 参数

2. 图形文件

(1) 图形界面坐标系。显示器屏幕是由像素点矩形阵列组成的，改变不同位置像素点的颜色即可实现绘制不同的图形。EGE 图形界面坐标系如图 1-21 所示，每个方格表示一个像素。水平向右是 x 轴正方向，垂直向下是 y 轴正方向。若显示图形界面宽 (width) 和高 (height) 皆为 10，则界面左上角第一个像素点坐标是 (0,0)，右下角最后一个像素点坐标是 (9,9)，屏幕范围内的每一个点都可以用坐标 (x,y) 表示，x 和 y 的范围都为 0~9。



图形文件-qzptlj

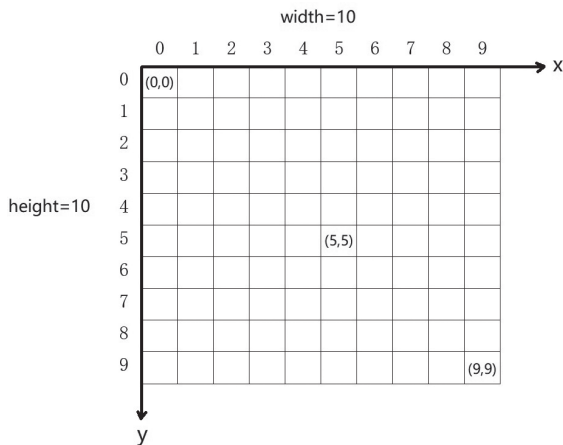


图 1-21 图形界面坐标系

(2) 常用图形操作函数。EGE 常用图形函数如表 1-1 所示。

表 1-1 EGE 常用图形函数

序号	函 数	功 能	举 例
1	initgraph(宽, 高);	进入并设置图形界面尺寸	initgraph(640,480);
2	closegraph();	关闭图形窗口	closegraph();





续表

序号	函 数	功 能	举 例
3	setbkcolor(颜色);	设置图形界面背景颜色	setbkcolor(WHITE);
4	setcolor(颜色);	设置图形前景 / 边框颜色	setcolor(RED);
5	setfillcolor(颜色);	设置图形填充颜色	setfillcolor(GREEN);
6	newimage();	初始化图片变量	PIMAGE img; img=newimage();
7	getimage(变量, " 图片名 ", 宽, 高);	载入图片	getimage(img, "1.jpg");
8	putimage(x,y, 变量);	在 (x,y) 位置显示变量中装载的图片	putimage(200,200,img);
9	putimage_withalpha(NULL, 变量,x1,y1, 宽 1, 高 1,x2,y2, 宽 2, 高 2);	显示透明背景图片 (x1,y1) 图片显示位置坐标 (宽 1, 高 1) 图片显示时尺寸 (x2,y2) 在原图片截取位置的左上角坐标 (宽 2, 高 2) 在原图片从左上角 (x2,y2) 位置 开始截取宽 2 和高 2 的矩形尺寸	putimage_withalpha(NULL, mali,mali_x,mali_y,140, 140,0,0,140,140);
10	delimage(变量);	删除图片	delimage(img);
11	putpixel(x,y, 颜色);	在图形界面 (x,y) 位置绘制像素点	putpixel(100,100,RED);
12	line(x1,y1,x2,y2);	画从 (x1,y1) 到 (x2,y2) 直线	line(10,10,100,100);
13	rectangle(x1,y1,x2,y2);	画从 (x1,y1) 到 (x2,y2) 为对角线的空心矩形 边线颜色由 setcolor() 函数指定	rectangle(50,50,150,150);
14	bar(x1,y1,x2,y2);	画从 (x1,y1) 到 (x2,y2) 为对角线的实心矩形 边线颜色由 setcolor() 函数指定 填充颜色由 setfillcolor() 函数指定	bar(50,50,150,150);
15	circle(x,y,r);	在 (x,y) 位置绘制半径为 r 的空心圆 边线颜色由 setcolor() 函数指定	circle(100,100,10);
16	fillellipse(x,y,w,h);	在 (x,y) 位置绘制实心椭圆, 当 w 等于 h 时 绘制正圆 边线颜色由 setcolor() 函数指定 填充颜色由 setfillcolor() 函数指定	setcolor(YELLOW); setfillcolor(BLUE); fillcircle(100,100,10);
17	outtextxy(x,y, " 字符串 ");	在图形界面 (x,y) 位置显示字符串	outtextxy(30,30, " 欢迎使用 EGE ! ");
18	xyprintf(x,y, " 格式符 ", 输 出量);	在图形界面 (x,y) 位置显示输出量的值	xyprintf(200,200,"%d",2+3);



注意

C 语言图形界面通过图片的左上角坐标定位图片的显示位置，如图 1-22 所示位置显示图片，其语句为 `putimage(10,10,fj);`。



图 1-22 确定图片显示位置

1.2.3 练一练

序 号	程 序 段	执行过程和结果
1	<pre>initgraph(300,300); circle(150,150,100); line(50,150,250,150); putpixel(150,150,RED); xyprintf(70,120,"%c=%d",'r',100);</pre>	
2	<pre>initgraph(600, 600); setcolor(RED); circle(300,300,300); setcolor(BLUE); circle(300,225,230); setcolor(GREEN); circle(300,175,175); setcolor(YELLOW); circle(300,125,125);</pre>	

拓展

编写程序，在 Dev-C++ 编辑器中验证 1.2.3【练一练】中显示的图形。





1.2.4 做一做

在屏幕横向相邻位置显示两张图片，如图 1-23 所示。



图 1-23 程序执行效果

【算法引导】

- (1) 显示两张不同的图片需要定义两个图片变量 f1 和 f2，用 getimage() 函数分别给两个变量载入不同的两张图片。
- (2) 确定好两张图片的尺寸和显示位置，可参考表 1-2 所示数据。

表 1-2 两张图片参考数据

图 片	宽	高	坐 标
1	300	180	0,0
2	300	180	300,0

思考

表 1-2 给出了两个图片的尺寸和显示位置，则显示界面尺寸应该设置为多少？

【算法流程】

行 号	算法流程
1	图形头文件
2	主函数首部
3	{
4	用 PIMAGE 数据类型声明图形变量 f1 和 f2
5	用 initgraph() 函数设置显示界面的宽和高
6	用 newimage() 函数初始化 f1
7	用 newimage() 函数初始化 f2
8	用 getimage() 函数给变量 f1 载入图片 1



续表

行 号	算法流程
9	用 <code>getimage()</code> 函数给变量 <code>fj2</code> 载入图片 2
10	用 <code>putimage()</code> 函数在指定坐标位置显示图片 1
11	用 <code>putimage()</code> 函数在指定坐标位置显示图片 2
12	暂停程序执行
13	返回值 0
14	}

根据上述算法流程给出该程序中关键的第 6 行、第 8 行和第 10 行的参考代码，如代码 1-5 所示。

代码 1-5

行 号	代 码
6	<code>fj1=newimage();</code>
8	<code>getimage(fj1,"fengjing-1.jpg");</code>
10	<code>putimage(0,0,fj1);</code>

【算法实现】

参考“算法引导”“算法流程”及代码 1-5，自主编写、调试、执行代码，锻炼学习思维和动手操作能力，执行结果正确后把正确代码写到下方。

思考

如何在“算法实现”中的代码的基础上，给图片添加图 1-25 所示的边框？



项目实现

根据项目描述中说明的功能实现方法和关键参考代码 1-6，再结合本项目中所学习的 C 语言程序基本结构和图形文件的知识，实现本项目中显示小鸡吃虫子动画背景图片的功能。



显示背景-4vefbj





代码 1-6

行 号	代 码	功 能
1	initgraph(1000,500);	根据背景图片的大小定义图形界面尺寸
2	PIMAGE bj;	定义存储背景图片变量 bj
3	bj=newimage();	初始化变量
4	getimage(bj,"bj.jpg");	给变量 bj 载入图片 bj.jpg
5	putimage(0,0,bj);	在显示界面左上角 (0,0) 坐标显示背景图片
6	getch();	暂停程序执行
7	closegraph();	关闭图形模式

【编写、调试代码】

写出调试成功的项目代码，并记录调试过程中的出错信息及修改情况。

代 码	
出 错 信 息 及 修 改 情 况	

**项目延伸**

如何实现在背景图片的基础上显示小鸡图片？直接用 putimage() 函数是否能达到预期效果？如果不能，应该如何修改？把自己的想法或者程序写在下方。



评价反馈

学生和教师根据表 1-3 所列各项内容考核评价项目 1 的学习情况。

表 1-3 学习情况考核评价表

班级		姓名	
日期		项目名称	
评价内容		学生自评（0~5 分）	教师评价（0~5 分）
知识技能	了解 C 语言		
	程序结构		
	图形坐标系		
	图形函数		
	图片函数		
操作技能	使用编辑器		
	编写程序		
	调试程序		
自主学习	项目准备		
	相关知识		
创新能力	项目思考		
	项目延伸		
素质提升	学习能力		
	协作能力		
思政提升	谦虚谨慎		
	自信自强		
总分	160 分		

巩固练习

一、选择题

1. C 语言是（ ）语言。
A. 机器 B. 汇编 C. 低级 D. 高级
 2. 计算机高档语言程序的运营措施有编译执行和解释执行两种，以下论述中正确的是（ ）。
- [2022 年 3 月计算机等级考试（二级 C 语言）考题]
- A. C 语言程序仅可以编译执行





- B. C 语言程序仅可以解释执行
 - C. C 语言程序既可以编译执行又可以解释执行
 - D. 以上说法都不对
3. 以下论述中错误的是 ()。[2022 年 3 月计算机等级考试 (二级 C 语言) 考题]
- A. C 语言的可执行程序是由一系列机器指令构成的
 - B. 用 C 语言编写的源程序不能直接在计算机上运行
 - C. 通过编译得到的二进制目的程序需要连接才可以运行
 - D. 在没有安装 C 语言集成开发环境的机器上不能运行 C 源程序生成的 .exe 文件
4. 以下叙述中正确的是 ()。[2019 年 9 月计算机等级考试 (二级 C 语言) 考题]
- A. C 语言比其他语言高级
 - B. C 语言可以不用编译就能被计算机识别执行
 - C. C 语言以接近英语国家的自然语言和数学语言作为语言的表达形式
 - D. C 语言出现得最晚, 具有其他语言的一切优点
5. 以下叙述中错误的是 ()。[2018 年 3 月计算机等级考试 (二级 C 语言) 考题]
- A. C 程序经过编译、连接步骤之后才能形成一个真正可执行的二进制机器指令文件
 - B. 用 C 语言编写的程序称为源程序, 它以 ASCII 代码形式存放在一个文本文件中
 - C. C 语言中的每条可执行语句和非执行语句最终都将被转换成二进制的机器指令
 - D. C 语言源程序经编译后生成后缀为 .obj 的目标程序

二、填空题

1. 计算机编程语言分为_____语言、_____语言和_____语言。
2. C 语言源程序文件名的后缀是_____, 目标文件名的后缀是_____, 可执行文件的后缀是_____。
3. C 语言程序由_____组成。
4. 在坐标 (10,10) 位置绘制一个宽为 300、高为 200 矩形的 C 语言语句是_____。
5. 以坐标 (300,300) 为圆心绘制两个半径分别为 300,150 的同心圆的 C 语言语句是_____。

项目 2

“显示动画中的小鸡” 的设计与实现

项目要求

知识要求

- ◎ 掌握 C 语言中的数据类型。
- ◎ 掌握算术运算符、位运算符和特殊运算符。
- ◎ 掌握常量和变量。
- ◎ 掌握格式输入输出函数。
- ◎ 掌握字符输入输出函数。
- ◎ 掌握顺序结构。

技能要求

- ◎ 熟练使用 C 语言编辑器编写、调试和执行程序。
- ◎ 会使用顺序结构编写简单程序。

思政要求

- ◎ 具有守正创新的精神。
- ◎ 具有踔厉奋发的精神。



项目描述

本项目实现“小鸡吃虫子”动画的第二步——在项目 1 中显示动画的背景图片上显示静止的小鸡，如图 2-1 所示。

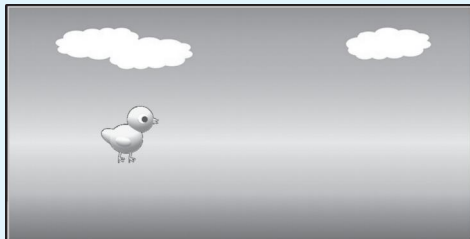


图 2-1 显示小鸡

实现方法：

1. 显示小鸡图片

小鸡图片是方形的，若想让小鸡图片更好地融入背景中，需要用到背景透明的 png 格式图片，结合 `putimage_withalpha()` 函数来实现。

函数格式：

```
putimage_withalpha(NULL, 变量, x1, y1, 宽 1, 高 1, x2, x2, 宽 2, 高 2);
```

参数说明：

- NULL：图片显示在屏幕上；
- 变量：存储 png 图片的变量；
- (x1,y1)：图片在屏幕上显示位置坐标（图片左上角位置）；
- (宽 1, 高 1)：图片在屏幕上显示尺寸；
- (x2,x2)：在 png 图片截取矩形区域的左上角坐标；
- (宽 2, 高 2)：在 png 图片从左上角 (x2, x2) 位置开始截取的宽 2 和高 2 的矩形区域。

2. 合理排序语句

本项目显示效果是背景图片在后，小鸡图片在前；即先显示背景图片，后显示小鸡图片；若相反，背景图片会把先显示的小鸡图片遮盖住。若想达到预期效果，就需要按照 C 语言中程序顺序结构的语法规则合理排序各种功能语句。

实现以上两个功能，需要用到 C 语言的图形文件和程序基本结构的相关知识。

因此在本项目中学习：

- (1) C 语言中常量的数据类型及各自的表现形式；
- (2) C 语言中变量的数据类型、定义和初始化方式；
- (3) C 语言中算术运算符、位运算符及其他特殊运算符；
- (4) C 语言中输入输出函数；
- (5) C 语言中程序编写的基本结构——顺序结构。



项目准备

根据【项目描述】中讲述的本项目中需要掌握的知识点，通过个人学习、小组讨论等方式完成以下预习任务：

(1) 写出什么是常量，并举例说明有哪些数据类型的常量。

(2) 写出什么是变量，有哪些数据类型的变量，并举例说明定义变量的格式。

(3) 写出常见的输入输出函数，以及其格式和功能。

(4) 写出 C 语言中的三大结构，以及各自的使用情景。





相关知识



2.1 格式输入输出

2.1.1 引导案例

求圆的周长和面积。

【案例分析】

- (1) 输入圆半径 r 的值求得圆的周长 c 和面积 s 并输出, r, c, s 都设置为浮点型。
- (2) 用 `scanf()` 格式输入函数给半径 r 输入值。
- (3) 输出圆的周长 c 和面积 s , 有两种处理方法。
 - ①用 `printf()` 格式输出函数直接输出计算周长和面积的表达式。
 - ②先把计算周长和面积的表达式分别赋值给 c 和 s 这两个变量, 再用 `printf()` 函数输出 c 和 s 。

【操作步骤】

- (1) 打开 Dev-C++ 编辑器, 执行“文件”→“新建”→“源代码”命令, 创建一个新的 C 语言程序源文件; 或者单击工具栏中的“新建”按钮, 或者按快捷键 `Ctrl+N` 也可以创建程序源文件。
- (2) 在代码编辑区输入代码 2-1。

代码 2-1

行 号	代 码	功 能
1	<code>#include<stdio.h></code>	头文件
2	<code>#define PI 3.14</code>	符号常量
3	<code>int main()</code>	函数首部
4	<code>{</code>	函数体开始
5	<code>float r=2.3,c,s;</code>	声明变量
6	<code>printf(" 输入半径 r: ");</code>	输入提示
7	<code>scanf("%f",&r);</code>	输入 r 的值
8	<code>c=2*PI*r;</code>	计算周长
9	<code>s=PI*r*r;</code>	计算面积
10	<code>printf(" 周长 =%8.2f\n",c);</code>	输出周长
11	<code>printf(" 面积 =%8.2f\n",s);</code>	输出面积
12	<code>return 0;</code>	返回值 0
13	<code>}</code>	函数体结束

- (3) 单击工具栏中的“编译运行”按钮运行代码, 若有错误, 修改后再次运行, 直至代码能

正确运行。程序执行结果如图 2-2 所示。

```
输入半径r: 3.5
周长= 21.98
面积= 38.47
```

图 2-2 程序执行结果

2.1.2 知识点解析

1. 常量

数值不变的量为常量。用来给变量赋值或者参与计算。C 语言中常量数据类型如表 2-1 所示。

表 2-1 常量数据类型

类 型 名	举 例	说 明
(短)整型	12	十进制
	012 对应十进制数 10	八进制，数字前加‘0’（数字零）
	0x12 对应十进制数 18	十六进制，数字前加‘0x’或‘0X’
长整型	12L, 012l	数字后面添加 L 或 l（小写字母 l）
浮点型（实型）	1.2	小数型
	1.2E3	指数型
字符型	'a','l','*'	一般字符，用单引号' '括起来
	'\n'（回车换行）	转义字符，用单引号' '括起来
	'\0'（空字符）	
	'\"'（双引号）	
	'\'（反斜杠）	
	'\ddd'（1-3 位八进制代表的字符）	
字符串	"abc","1234"	用双引号" "括起来
	#define PI 3.14	#define 符号常量名 常量值

注意

定义符号常量是一条预处理命令，不是一条语句，最后不能加分号。

2. 变量

变量是在程序运行过程中其值可改变的量。在 C 语言中变量要先声明其数据类型后才能使用，声明变量的目的是让系统给变量分配一个合理的存储空间。C 语言中常用变量数据类型及所占字节如表 2-2 所示。



变量-2pq0vt

表 2-2 变量数据类型及所占字节

类 型 名	标 识 符	占字节数
字符型	char	1
(短)整型	int	2
长整型	long	4
单精度浮点型	float	4
双精度浮点型	double	8



声明变量格式:

数据类型 变量名 1, 变量名 2...;

【例 2-1】 定义一个（短）整型变量 a，一个单精度浮点型变量 b。

```
int a;  
float b;
```

变量的名字由字母、数字和下划线组成，并由字母和下划线开头。

【例 2-2】 变量名命名举例。

a_ 和 _a 和 a_3 是合法的变量名；3_a 和 a*3 是不合法的变量名。

注意

C 语言中是区分大小写字母的，如 A_3 和 a_3 是两个变量名。

在定义变量的同时可以给变量赋值，称为变量的初始化。赋值符号是“=”，功能是把符号右边的数值或表达式结果赋给符号前面的变量。

【例 2-3】 定义一个（短）整型变量 a 和 b，分别赋值为 10 和 20。

```
int a=10,b=20;
```

一种数据类型一次性定义多个变量时，变量间用逗号隔开；定义语句后可同时给这多个变量赋相同值。

【例 2-4】 定义一个（短）整型变量 a 和 b，都赋值为 10。

```
int a,b;  
a=b=10;
```

注意

语句 int a=b=10; 是错误的。

3. 算术运算符

C 语言中有算术运算符、关系运算符、逻辑运算符和位运算符等，本项目中讲解算术运算符和位运算符，关系运算符和逻辑运算符在项目 3 中讲解。C 语言中常用的算术运算符如表 2-3 所示。

表 2-3 常用算术运算符

运 算 符	名 称	功 能	举 例
+	加	两个数相加	$13+2 \Rightarrow 15$
-	减	两个数相减	$13-2 \Rightarrow 11$
*	乘	两个数相乘	$13*2 \Rightarrow 26$



续表

运 算 符	名 称	功 能	举 例
/	除	两个数相除 (当两个数均为整数时, 结果为整数, 小数位全部舍掉)	$13/2 \Rightarrow 6$ $13.0/2 \Rightarrow 6.500\ 000$
%	求余	两个整数相除的余数	$13\%2 \Rightarrow 1$

【例 2-5】分析下面程序段的执行结果。

```
int n=123,i,j;
i=n/100;    //n 和 100 都是整型, 结果也是整型, 所以 i 的值为 1
j=n%10;     //n 对 100 求余, 所以 j 的值为 3
printf("i=%d,j=%d",i,j);
执行结果: i=1,j=3
```

思考

如何取出【例 2-5】中数字 123 的十位数 2?

4. 位运算符

计算机系统的内存由若干字节 (byte) 组成的, 1 字节由 8 个二进制位 (bit) 构成, 每位取值 0 或 1, 位运算符就是计算机对每位的二进制数进行运算。若程序中参与计算的不是二进制数, 系统会自动转换成二进制数再进行按位操作。C 语言中位运算符如表 2-4 所示。

表 2-4 位运算符

运 算 符	名 称	功 能	举 例
&	按位与	两个位都为 1 时, 结果为 1	$(0101)_2 \& (0011)_2 \Rightarrow (0001)_2$
	按位或	两个位都为 0 时, 结果为 0	$(0101)_2 (0011)_2 \Rightarrow (0111)_2$
^	按位异或	两个位相同结果为 0, 相异结果为 1	$(0101)_2 \wedge (0011)_2 \Rightarrow (0110)_2$
~	按位取反	0 变 1, 1 变 0	$\sim(0101)_2 \Rightarrow (1010)_2$
<<	左移	二进制全部左移若干位, 高位丢弃, 低位补 0	$(0101)_2 << 2 \Rightarrow (0100)_2$
>>	右移	二进制全部右移若干位, 无符号数, 高位补 0 有符号数, 根据编译器处理方法, 有的补符号位 (算术右移), 有的补 0 (逻辑右移)	$(0101)_2 >> 2 \Rightarrow (0001)_2$

注意

1. 位运算的操作数只能是整型或字符型, 运算结果是整型。
2. 位运算的操作数按二进制补码形式参与运算。
3. 位运算符优先级从高到低为 “~” → “<<” 或 “>>” → “&” → “^” → “|”。



**【例 2-6】** 分析下面程序段的执行结果。

[2021 年 3 月计算机等级考试 (二级 C 语言) 考题]

```
int c;
c=13|5; // (13)10 ⇨ (1101)2, (5)10 ⇨ (0101)2, 两数按位或后结果为 (1101)2
printf("%d\n",c);
```

执行结果: 13

【例 2-7】 分析下面程序段的执行结果。

[2022 年 3 月计算机等级考试 (二级 C 语言) 考题]

```
unsigned char a=8,c;
c=a>>3; // (8)10 ⇨ (1000)2, 右移 3 位后为 (0001)2 ⇨ (1)10
printf("%d\n",c);
```

执行结果: 1

【例 2-8】 分析下面程序段的执行结果。

[2020 年 3 月计算机等级考试 (二级 C 语言) 考题]

```
int a=1,b=2,c=3,x; // (1)10 ⇨ (0001)2, (2)10 ⇨ (0010)2, (3)10 ⇨ (0011)2
x=(a^b)&c; // (0001)2 ^ (0010)2 ⇨ (0011)2, (0011)2 & (0011)2 ⇨ (0011)2
printf("%d\n",x);
```

执行结果: 3

5. 特殊运算符

在 C 语言中还有其他特殊形式的运算符, 可以简化表达式的书写, 如表 2-5 所示。

表 2-5 特殊形式运算符

特殊运算符	名 称	功 能	举 例
++	自增	i++; 或 ++i; 等价于 i=i+1;	i=13;i++; ⇨ i=14
--	自减	i--; 或 --i; 等价于 i=i-1;	i=13;i--; ⇨ i=12
+=	加赋值	i+=j; 等价于 i=i+j;	i=13;j=2; i+=j; ⇨ i=i+j; ⇨ i=13+2=15
-=	减赋值	i-=j; 等价于 i=i-j;	i=13;j=2; i-=j; ⇨ i=i-j; ⇨ i=13-2=11
=	乘赋值	i=j; 等价于 i=i*j;	i=13;j=2; i*=j; ⇨ i=i*j; ⇨ i=13*2=26
/=	除赋值	i/=j; 等价于 i=i/j;	i=13;j=2; i/=j; ⇨ i=i/j; ⇨ i=13/2=6
%=	求余赋值	i%=j; 等价于 i=i%j;	i=13;j=2; i%=j; ⇨ i=i%j; ⇨ i=13%2=1



续表

特殊运算符	名 称	功 能	举 例
&=	位与赋值	$i \&= j$; 等价于 $i = i \& j$;	$i = 13; j = 2$; $i \&= j; \Rightarrow i = i \& j; \Rightarrow i = 13 \& 2 = 0$
=	位或赋值	$i = j$; 等价于 $i = i j$;	$i = 13; j = 2$; $i = j; \Rightarrow i = i j; \Rightarrow i = 13 2 = 15$
^=	位异或赋值	$i ^= j$; 等价于 $i = i ^ j$;	$i = 13; j = 2$; $i ^= j; \Rightarrow i = i ^ j; \Rightarrow i = 13 ^ 2 = 15$
<<=	左移赋值	$i <<= j$; 等价于 $i = i << j$;	$i = 13; j = 2$; $i <<= j; \Rightarrow i = i << j; \Rightarrow i = 13 << 2 = 52$
>>=	右移赋值	$i >>= j$; 等价于 $i = i >> j$;	$i = 13; j = 2$; $i >>= j; \Rightarrow i = i >> j; \Rightarrow i = 13 >> 2 = 3$
,	逗号运算符	表达式 1, 表达式 2, ..., 表达式 n; 最后一个表达式的结果是整个逗号表达式的值	$i = 13; j = 2$; $j++, i += j, i \% = j; \Rightarrow i = 1 \quad j = 3$ 整个逗号表达式结果为 1

特殊强调符号“++”和“--”的使用。这两个符号单独使用对变量自增和自减，即 $i++$ （后加）和 $++i$ （前加）两条语句的功能是相同的，都是对变量 i 在原有数值的基础上加 1，表达式结果相同。但是如果放在其他表达式或语句中， $i++$ 和 $++i$ 会影响表达式或语句的结果。

【例 2-9】 分析下面程序段的执行结果。

程序段 1:

```
int i=3, j=6, a, b;
printf("i=%d, j=%d\n", i++, ++j); // 先输出 i 值 3 后再加 1, i 值为 3
// j 值 6 加 1 后再输出, j 值为 7
i=3, j=6;
a=i--; // 先把 i 值 3 赋给 a 后再减 1, i 值为 2, a 值为 3
b=--j; // 先把 j 值 6 减 1 后再赋给 b, j 值为 5, a 值为 5
printf("i=%d, j=%d, a=%d, b=%d", i, j, a, b);
```

执行结果:

```
i=3, j=7
i=2, j=5, a=3, b=5
```

程序段 2:

```
int i=3, j=6, a, b; // "=" 赋值优先级高于 "," 逗号
a=++i, j--; // 先把 i 值 3 加 1 后再给 a 赋值, i 值为 4, a 值为 4
// 最后计算 j, j 值 6 减 1, j 值为 5
printf("i=%d, j=%d, a=%d\n", i, j, a);
i=3, j=6;
```





```
b=i++,--j; // 先把 i 值 3 赋值给 b 后再加 1, i 值为 4, b 值为 3
           // 最后计算 j, j 值 6 减 1, j 值为 5
printf("i=%d,j=%d,b=%d",i,j,b);
执行结果:
i=4,j=5,a=4
i=4,j=5,b=3
```

注意

(1) 求余符号 "%" 的两边操作数一定是整型。如果不是整型可以强制转换, 格式为“(数据类型)表达式”。如 (int)3.14 结果为 3, 小数位全部舍掉。也可以把整型转换成浮点型, 如 (float)3 结果为 3.000000。

(2) 两数做算术操作时, 如果数据类型不同, 结果扩展为占字节数大的类型。例如, 13.0+2 的结果为 15.00000。(在 C 语言中输出浮点型数值时系统默认小数位为 6 位)

(3) C 语言中表达式中所有符号都不能省略。例如, $s=ab$; 写成 $s=a*b$; $v=\frac{a}{b}$; 写成 $v=a/b$;

6. 格式输入输出

格式输入输出函数用来指定输入和输出数据的格式, 使输入和输出样式规范化。

(1) 格式输入函数。其语句格式如下:

```
scanf(" 格式符 ", & 输入量 );
```



格式输入输出-4hjp3y

执行 scanf() 函数时执行界面会出现闪烁的光标, 等待用户通过键盘给输入量随机输入数值。不同数据类型的输入量对应不同的“格式符”, 参见表 2-6。符号“&”是取地址运算符。

表 2-6 格式符及其功能

格 式 符	功 能
%d	十进制整数
%ld	十进制长整型数
%u	无符号十进制整数
%o	八进制整数
%x 或 %X	十六进制整数
%f	浮点数
%e 或 %E	浮点数的指数形式
%g 或 %G	系统自动选择 %f 或 %e 格式, 选择长度较短格式
%c	单个字符
%s	单个字符串