

产教融合协同育人项目成果

# 程序逻辑与思维

## (C语言)

罗勇 张祎 主编

吕波 苏绍培 副主编

张国胜 王俊海

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书是校企深度融合的产物,是由企业高级软件工程师和高校教学经验丰富的教师共同打造的新型工作手册式教材。本书打破传统教材的章节体系,对所有知识体系进行了重构,对实际项目生产过程进行了设计。

本书涵盖 C 语言的所有知识体系,共有 9 个模块,其内容分为 3 个阶段:第 1 阶段(模块 1 和模块 2)为马步(打基础),主要介绍 C 语言的基础知识,包含常量与变量、运算符与表达式、程序输入与输出等内容;第 2 阶段(模块 3~8)为专项(分内容),主要对分支结构、循环结构、数组、算法、函数、结构体、文件操作等内容进行专项介绍;第 3 阶段(模块 9)为作战(成团队),主要通过对完整的软件项目开发流程进行设计,完成项目的设计、开发、文档、交付等内容。

本书的任务内容以“开门见山”的模式进行设计,即目标描述、接领任务、分析任务、制定方案、实施实现、测试验收、总结拓展,以更好地培养初学者的程序逻辑与思维能力,更顺利地开启软件世界的“第一扇门”,为后续的学习奠定扎实的“根基”。

本书可作为高校 ICT 专业的 C 语言程序设计课程的教学教材,重点培养其程序逻辑与思维能力,也可作为软件编程入门者的学习用书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

程序逻辑与思维: C 语言 / 罗勇, 张祎主编. —北京: 电子工业出版社, 2021.9  
ISBN 978-7-121-41995-9

I. ①程… II. ①罗… ②张… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2021)第 187631 号

责任编辑: 徐建军 文字编辑: 王 炜

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 17 字数: 435.2 千字

版 次: 2021 年 9 月第 1 版

印 次: 2021 年 9 月第 1 次印刷

印 数: 2 000 册 定价: 59.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010) 88254570, xujj@phei.com.cn。

# 前言

## Preface

伴随着我国经济转型的不断加速, ICT 作为产业经济结构转型的新动能, 其价值日益凸显, 行业保持较为快速的发展, 新一代信息技术已全面渗透到经济社会的各个领域, 改变着人们的生产、生活和思维方式, 成为推动经济发展的重要引擎。我国 ICT 产业却面临着人才供给绝对量缺少、人才错位、企业招不到合适的新人等问题, 严重制约着 ICT 产业的健康、快速发展。而程序基础类课程在整个 ICT 产业人才培养中起着举足轻重的作用, 因为它是引领初学者开启软件世界的“第一扇门”, 所以培养学生的程序逻辑与思维能力已刻不容缓。

市面上关于程序基础类的图书有很多, 大多数注重知识体系的编写, 而忽略了初学者的感受。作为引领初学者开启软件世界“第一扇门”的课程, 更应该注重培养初学者在解决实际问题时, 分析问题、确定思路, 以及选择相应的知识或技术的能力。

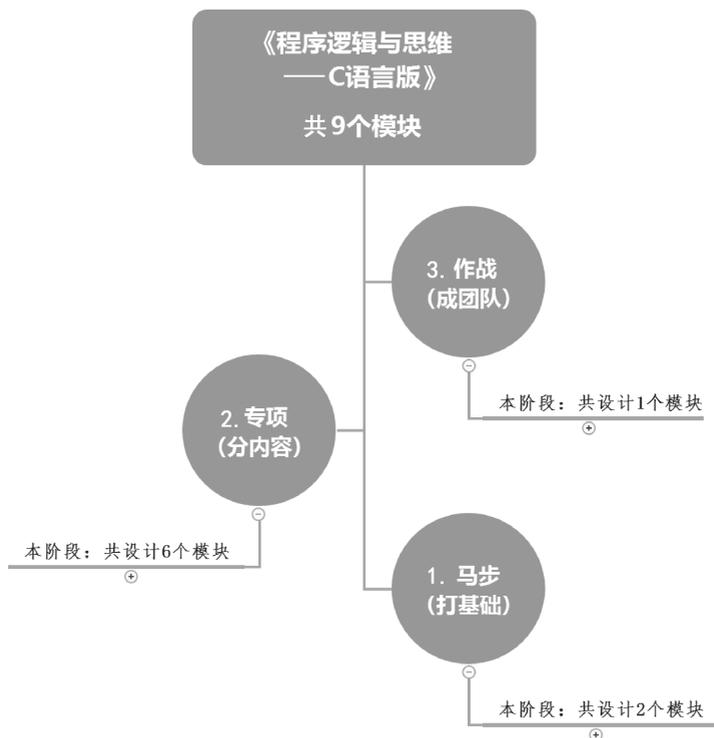
本书打破了传统教材的章节体系, 对所有知识体系进行了重构, 按实际项目生产过程进行设计。全书内容分为 3 个阶段, 共计 9 个模块 39 个任务, 基本涵盖 C 语言所有知识体系。第 1 阶段(模块 1 和模块 2)为马步(打基础), 主要介绍 C 语言的基础知识, 涉及常量与变量、运算符与表达式、程序输入与输出等内容; 第 2 阶段(模块 3~8)为专项(分内容), 主要对分支结构、循环结构、数组、算法、函数、结构体、文件操作等内容进行专项介绍; 第 3 阶段(模块 9)为作战(成团队), 主要通过对完整的软件项目开发流程进行设计, 完成项目的设计、开发、文档、交付等内容。

本书所有任务以“开门见山”的模式进行设计, 任务与任务之间以“故事情节”的方式进行设计, 让学习变得更加简单有趣。同时, 在每个任务中都结合技术知识选取一些人生哲理进行总结, 给枯燥的程序代码增加些许“温度”, 让学生更愿意接近和品研, 实现“带着自己的思想, 快乐学习”, 开启软件世界的“第一扇门”, 为后续的学习奠定扎实的“根基”。

本书具有以下特点。

### ■知识重构——去除传统课程的“教条”, 重构了知识体系

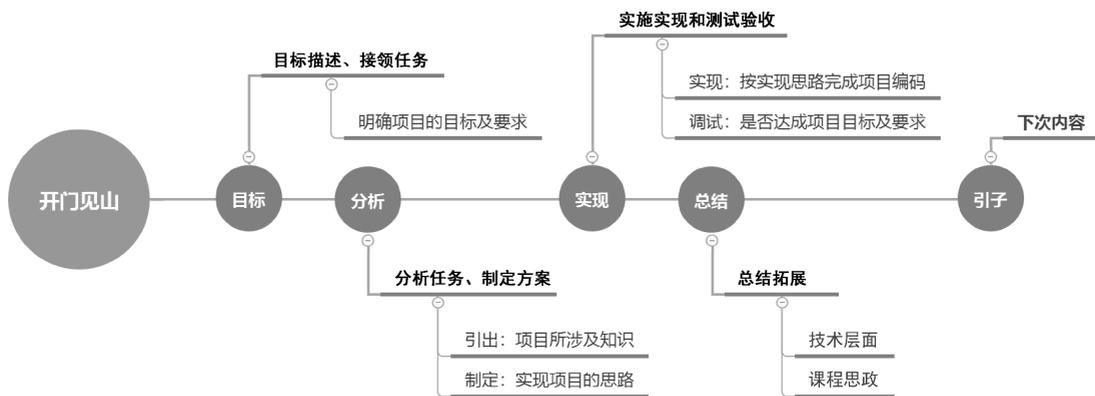
在内容设计上打破了传统说教的形式, 对所有知识体系进行了重构, 分为 3 个阶段, 共计 9 个模块, 基本涵盖 C 语言的所有知识体系。



■问题导向——提出问题，带着解决问题的思路去学习

所有任务都以“开门见山”的模式进行设计，即目标描述、接领任务、分析任务、制定方案、实施实现、测试验收和总结拓展，形成知识点的递进与复用关系。

任务与任务之间以“故事情节”的方式进行设计，让学习变得更加简单有趣。



■课程思政——春风化雨暗传课，思政育人细无声

坚持以“立德树人”作为教育的根本任务，将价值塑造、知识传授和能力培养三者融为一体，帮助初学者塑造正确的世界观、人生观、价值观。

本书由四川科技职业学院鼎利教育的罗勇和刘勇军组织编写，其中模块 1、模块 2、模块 7、模块 9 由罗勇、苏绍培、张国胜编写；模块 8 由四川长江职业学院李天祥编写；模块 3、模块 4、模块 5 由雅安职业技术学院的张祎、吕波、李琳、彭茜、杜锐和鼎利教育的刘勇军编写；模块 6 由雅安职业技术学院的王俊海编写。

本书在编写过程中，参阅了同类教材和文献资料，同时还引用了一些专家和企业家的理论和观点，并得到了广大师生的支持和帮助，在此表示衷心的感谢。

由于编者水平有限，书中难免存在不妥或错误之处，敬请广大读者批评指正。

为了更好地为读者提供教学资源，本书所有任务已录制教学视频，请扫二维码进行学习。另外，可直接与编者联系（E-mail: 13135166@qq.com）。



编 者



# 目录

## Contents

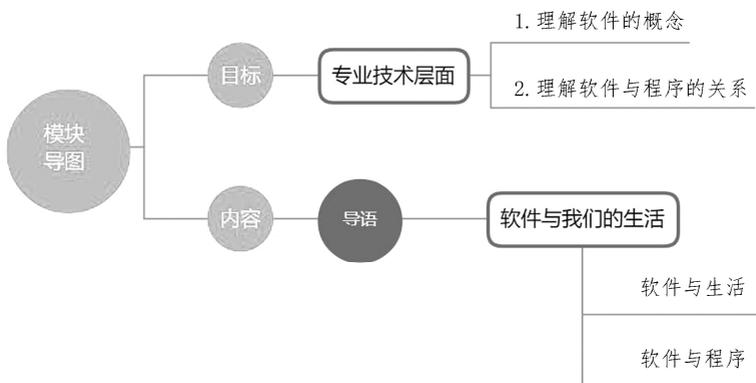
模块 1 软件与我们的生活	(1)
1.1 软件与生活	(2)
1.2 软件与程序	(4)
模块 2 C 语言程序中的输入/输出	(5)
项目导语: Hello World!	(6)
任务 1 触摸程序的肌肤	(7)
任务 2 程序中数据的保存	(15)
任务 3 程序中数据的显示	(21)
任务 4 程序中数据的输入	(28)
任务 5 程序中数据的计算	(33)
模块 3 C 语言程序中的分支结构应用	(39)
项目导语: 程序和人生一样面临选择	(40)
任务 1 计算机如何知道数的大与小	(41)
任务 2 今天发工资了, 你该缴多少税	(46)
任务 3 “我”也可以做选择	(54)
任务 4 优化体质指数 (BMI) 程序	(61)
模块 4 C 语言程序中的循环结构处理	(67)
项目导语: 重复就是一种精益求精的过程	(68)
任务 1 5050 这个数还面熟吗	(69)
任务 2 奇偶数也好玩	(75)
任务 3 怪物在什么时候倒下	(80)
任务 4 多久才能存够 100 元	(86)
任务 5 一对“孪生兄弟”	(92)
模块 5 C 语言程序中的数组应用	(101)
项目导语: 蜜蜂的启示	(102)
任务 1 数据的逆序输出	(104)
任务 2 找到你的位置, 并排到队伍中去	(110)
任务 3 数据就像水泡一样从下往上冒出来	(115)
任务 4 试着把“杨辉三角”显示到屏幕上	(122)

<b>模块 6 C 语言程序中的算法应用</b> .....	(129)
项目导语: 解决问题之道 .....	(130)
任务 1 使用程序求解我国古代的“百钱百鸡”问题 .....	(131)
任务 2 使用程序求解“四叶玫瑰数”问题 .....	(137)
任务 3 使用程序求解“兔子繁殖”问题 .....	(141)
<b>模块 7 C 语言程序中的函数及结构体应用</b> .....	(148)
项目导语: 前奏真的很重要 .....	(149)
任务 1 由四大名著引出的思考 .....	(150)
任务 2 信息的反馈 .....	(158)
<b>模块 8 C 语言程序中的文件操作应用</b> .....	(169)
项目导语: “日出而作, 日入而息”的规律 .....	(170)
任务 1 找个地方休息一下 .....	(171)
任务 2 起床了都出来露个脸 .....	(179)
任务 3 与结构体和函数一起玩玩 .....	(186)
<b>模块 9 C 语言程序综合项目实施</b> .....	(201)
项目需求 .....	(202)
任务 1 项目分析与设计 .....	(203)
任务 2 数据存储层实现 .....	(212)
任务 3 业务逻辑层——添加枪械信息实现 .....	(219)
任务 4 业务逻辑层——加载枪械信息实现 .....	(223)
任务 5 业务逻辑层——修改枪械信息实现 .....	(226)
任务 6 业务逻辑层——删除枪械信息实现 .....	(230)
任务 7 表示层——显示枪械界面实现 .....	(233)
任务 8 表示层——添加枪械界面实现 .....	(237)
任务 9 表示层——修改枪械界面实现 .....	(241)
任务 10 表示层——删除枪械界面实现 .....	(245)
任务 11 表示层——项目主界面菜单实现 .....	(249)
任务 12 表示层——程序主函数实现 .....	(253)
任务 13 软件项目文档编写 .....	(257)



# 模块 1

## 软件与我们的生活



## 1.1 软件与生活



随着科学技术的发展，各行各业都离不开软件，下面我们就来介绍几款与日常生活息息相关的软件。



### 我们的出行

“互联网+”时代的出行，让行程更易掌握。

通过使用出行软件，我们可以知道周围有哪些出租车和共享单车，使出行变得更加高效，如图 1.1 和图 1.2 所示。



图 1.1 出行软件示意



图 1.2 共享单车示意

通过沟通软件，使人们的交流不再遥不可及，让地球成为“世界村”，如图 1.3 所示。



### 我们的沟通

“互联网+”时代的地球，已经不再遥远，地球成为“世界村”啦！

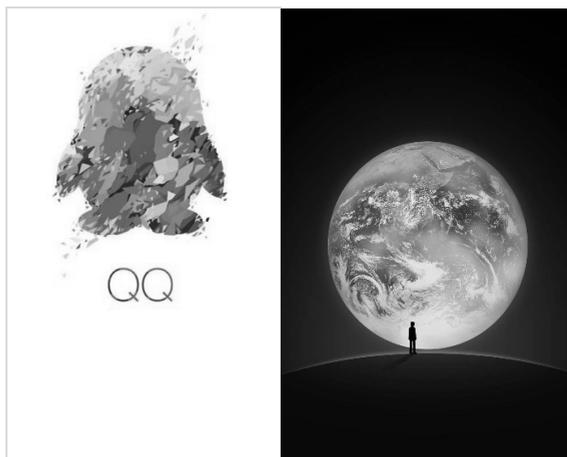


图 1.3 沟通软件示意

通过使用电子商务软件，让人们坐在家就可以购买心仪的商品，如图 1.4 所示。



## 我们的购物

“互联网+”时代的购物可以足不出户。



图 1.4 电商软件示意

在任何地点我们都可以利用碎片时间进行“充电”，如图 1.5 所示。



## 我们的学习

“互联网+”时代的学习可以随时开始。



图 1.5 在线教育软件示意

软件正在悄悄地改变着人们的生活，如图 1.6 所示。



## 各行各业都离不开软件

软件已渗透于我们生活的各个方面。



图 1.6 “互联网+”行业示意

软件正在改变着人们的生活及生活方式。在这个时代中，我们应充分享受时代赋予的便捷与快乐，同时更应思考该怎样创造出自己的价值。

## 1.2 软件与程序

上面给大家分享了生活中使用的一些相关软件，那么什么是软件呢？

软件就是按照特定顺序组织的计算机数据和指令的集合，软件还有以下几种定义。

### 1. 国标中对软件的定义

指与计算机系统操作有关的程序、规程、规则，以及可能有的文件、文档及数据。

### 2. 其他定义

(1) 指运行时，能够提供所要求功能和性能的指令或计算机程序集合。

(2) 指程序能够处理信息的数据结构。

(3) 指描述程序功能需求，以及程序如何操作和使用所要求的文档。

### 3. 以开发语言作为描述语言的定义

软件=程序+数据+文档。

以上简单地从专业角度对软件进行了介绍，那么软件是如何开发出来的呢？

软件是通过程序语言编写完成的，可以编写软件程序的语言如下。

(1) C 语言

(2) C++语言

(3) Java 语言

(4) Python 语言

(5) C#语言

(6) Objective-C/Swift 语言

.....

接下来，我们就基于 C 语言详细介绍软件程序的开发过程，带领初学者开启软件世界的“第一扇门”，为后续软件开发课程的学习奠定扎实的基础。

# 模块 2

## C 语言程序中的输入/输出

### 模块导图



## 项目导语：Hello World!



从这里开始，我们将开启程序编码生涯。

那么就从“Hello World!”开始吧！

不管学习哪一门程序语言，出现的第一个范例都是“Hello World!”，如图 2.1 所示。

<pre>C 1   #include &lt;stdio.h&gt; 2   int main() 3   { 4       printf("Hello, World!"); 5       return 0; 6   }</pre>	<pre>C# 1   namespace HelloWorld 2   { 3       class Program 4       { 5           static void Main(string[] args) 6           { 7               System.Console.WriteLine("Hello, World!"); 8           } 9       } 10   }</pre>
<pre>Java 1   public class HelloWorld 2   { 3       public static void main(String[] args) 4       { 5           System.out.println("Hello, World!"); 6       } 7   }</pre>	<pre>C++ 1   #include &lt;iostream&gt; 2   using namespace std; 3   int main() 4   { 5       cout &lt;&lt; "Hello, World!" &lt;&lt; endl; 6       system("pause"); 7       return 0; 8   }</pre>

图 2.1 程序语言中的“Hello World!”

“Hello World!”译为“你好世界!”，它是指编写程序在计算机屏幕上输出“Hello World!”的一行字符串。最初这个例程是在 Brian Kernighan 和 Dennis M. Ritchie 合著的 *The C Programming Language* 中使用的第一个演示程序，后来的程序员在学习编程或进行设备调试时便延续了这个习惯。

其实，程序中的“Hello World!”就是给所有程序员的第一个见面礼，寓意着向计算机语言世界问好，同时也意味着软件的世界中又多了一位成员。所以，“Hello World!”从某种意义上讲，就是要求我们带着一颗敬畏之心开始 C 语言程序的学习。



## 初识 C 语言程序

为了能够较全面地介绍 C 语言的相关知识，本模块共设计以下 5 个任务。

- 任务 1：触摸程序的肌肤。
- 任务 2：程序中数据的保存。
- 任务 3：程序中数据的显示。
- 任务 4：程序中数据的输入。
- 任务 5：程序中数据的计算。



## 任务 1 触摸程序的肌肤



### 目标描述

#### 任务描述

- **目标实现**  
正确安装配置 C 语言的开发环境，做好开发前的准备工作。
- **技术层面**  
理解 C 语言程序的执行流程。  
理解软件集成开发环境的意义。  
掌握 C 语言语法的基本规范。
- **课程思政**  
工欲善其事，必先利其器。

### 学习活动 1 接领任务

#### 领任务单

- **任务确认**  
本任务应完成以下内容。  
(1) 掌握 C 语言的基本语法规范。  
(2) 正确安装配置 C 语言集成的开发环境。
- **确认签字**

### 学习活动 2 分析任务

要完成本任务中的内容就必须学习以下知识。

## 知识学习：C 语言程序的集成开发环境执行流程和基本语法规范 知识学习

### 1. 集成开发环境

集成开发环境（Integrated Development Environment, IDE）是用于提供程序开发环境的应用程序。它包括代码编辑器、编译器、调试器和图形用户界面等工具。集成了代码的编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务组（套）。所有具备这个特性的软件或软件组都可称为集成开发环境。

#### 学习笔记

.....

.....

.....

那么用于开发 C 语言程序的集成开发环境有哪些呢？主要有以下 4 种。

#### (1) Turbo C

它是“爷爷辈”的 IDE，编译速度快，Turbo C 2.0 不支持鼠标，并且不能同时编辑多个文件。

#### (2) Visual C++ 6.0

它是经典开发工具，体积大（500MB），与流行操作系统有冲突。

#### (3) Dev C++

它的体积小（9MB），性能不友好，已于 2005 年后放弃更新了。

#### (4) Code::Blocks

它是目前如日中天的开源免费 C/C++ 开发工具，专业开发人员推荐使用。本书将以 Code::Blocks 作为开发工具。

### 2. C 语言程序的执行流程

集成开发环境是将程序代码的编写、分析、编译、调试等功能集成在一起，让开发者专注于代码的编写与实现，而将其他工作交由 IDE 自行完成，可大大提高开发效率。

但作为初学者，还是有必要了解 C 语言程序的真实执行流程的，如图 2.2 所示。

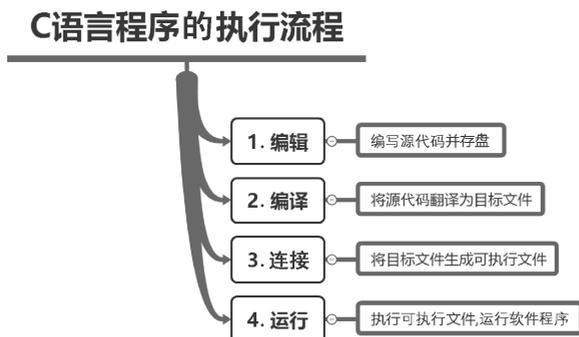


图 2.2 C 语言程序的执行流程示意

#### 说明：

一个 C 语言程序的执行共有 4 个阶段。

(1) 编辑：编写程序代码并存盘，这些代码就是源代码。

(2) 编译：将编写好的源代码翻译为目标文件。

(3) 连接：将目标文件生成可执行文件。

(4) 运行：执行可执行文件，运行软件程序。

因为，计算机只能识别二进制数，但编写的程序代码却不是二进制的，所以要经过以上步骤将编写的程序代码转化为计算机能识别的二进制数运行。

使用集成开发环境就可以将这 4 个过程集成在一起，我们只需要关心把程序代码编写好即可。



## 学习活动 4 实施实现

## 任务实现

## ● 实现代码

(1) 到官网下载 Code::Blocks 开发工具。

(2) 安装 Code::Blocks 开发工具。

① 找到安装文件，双击它启动安装，进入欢迎界面，如图 2.3 所示，单击“Next”按钮。



图 2.3 欢迎界面

② 进入版权许可界面，如图 2.4 所示，单击“I Agree”按钮。

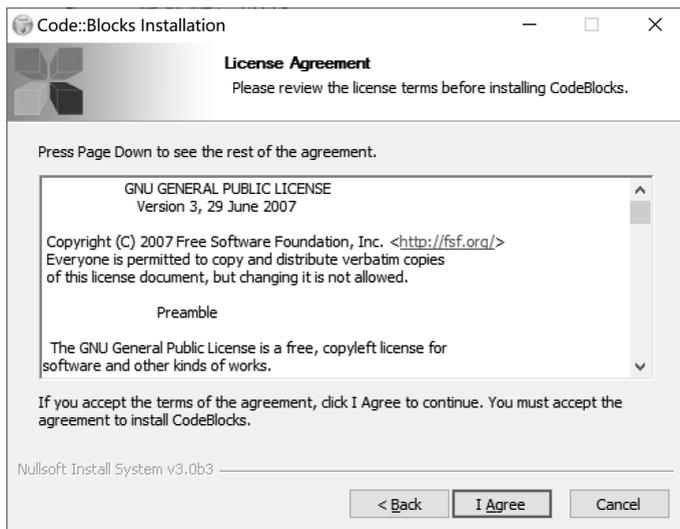


图 2.4 版权许可界面

③ 进入选择界面，如图 2.5 所示，单击“Next”按钮。

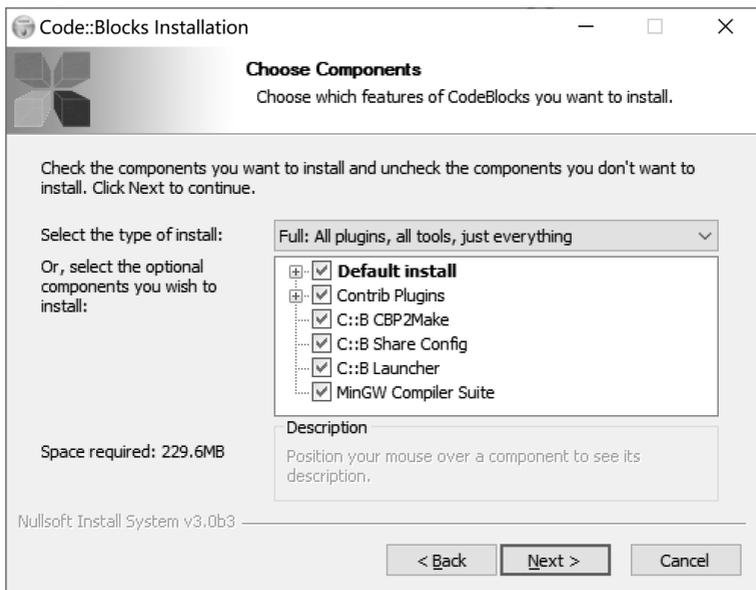


图 2.5 选择界面

④ 进入安装路径界面，如图 2.6 所示，单击“Install”按钮。

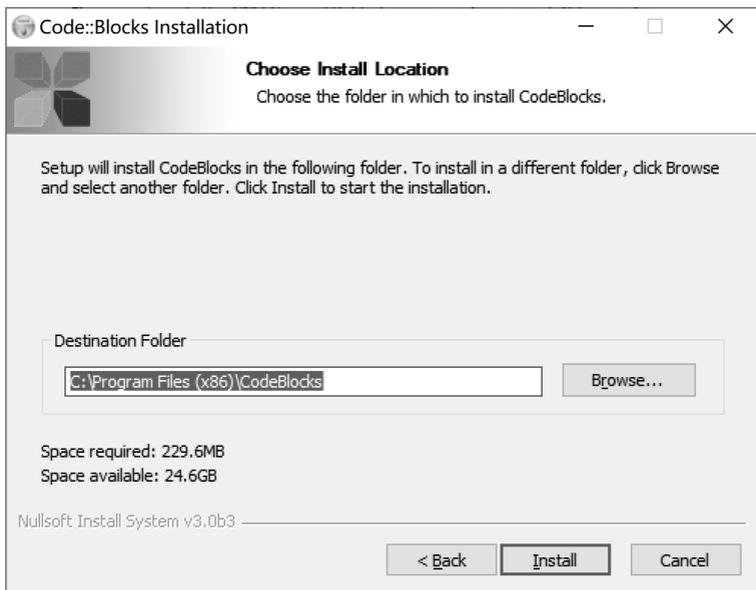


图 2.6 安装路径界面

⑤ 安装完成界面如图 2.7 所示。

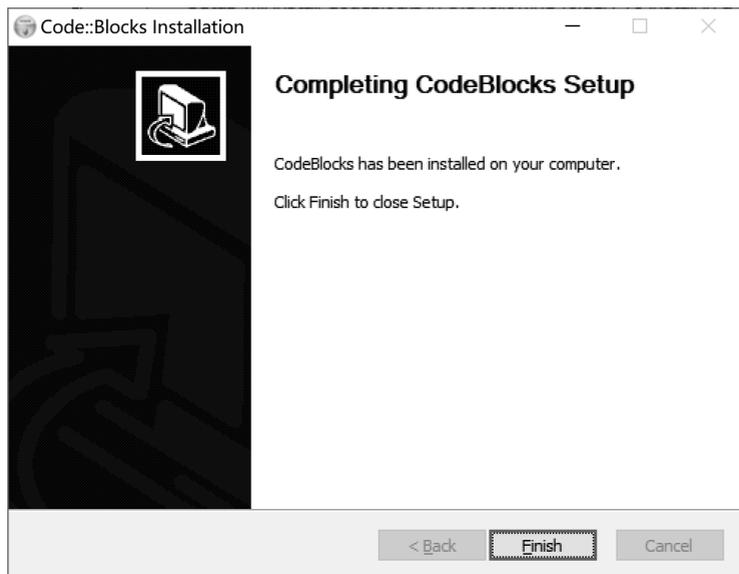


图 2.7 安装完成界面

(3) 创建 C 语言项目。

① 启动 Code::Blocks 软件。

② 执行“File”→“New”→“Project...”，如图 2.8 所示。

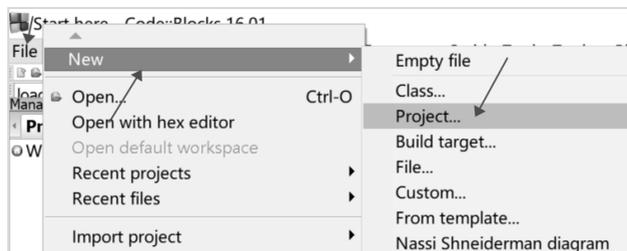


图 2.8 新建 C 语言项目

③ 选择项目类型为控制台，如图 2.9 所示。

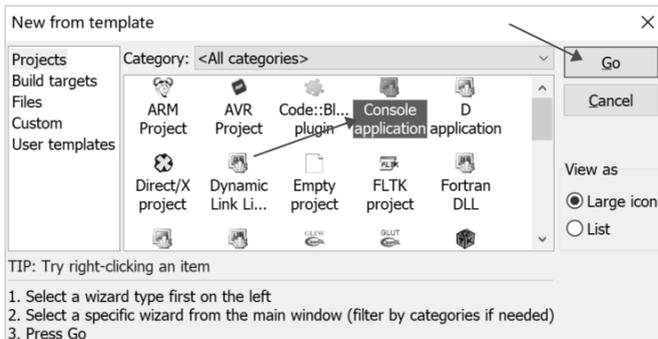


图 2.9 选择项目类型

- ④ 选择开发语言为 C 语言，单击“Next”按钮，如图 2.10 所示。

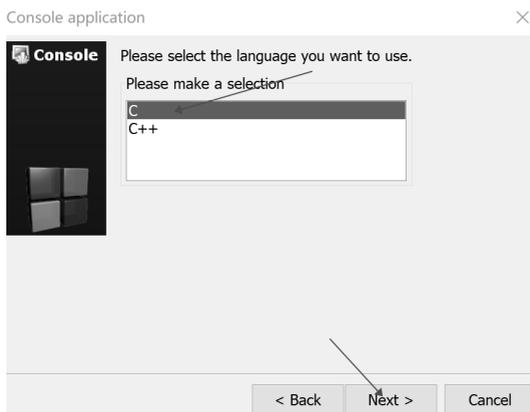


图 2.10 选择开发语言项目

- ⑤ 填写项目名称，选择项目保存位置，如图 2.11 所示。

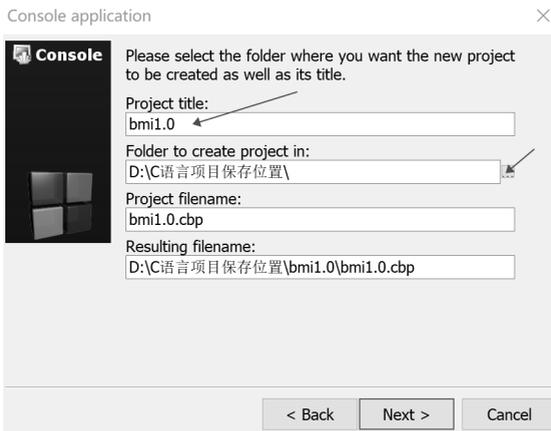


图 2.11 填写项目信息

- ⑥ 展开项目中的“main.c”文件，进入代码编写界面，如图 2.12 所示。

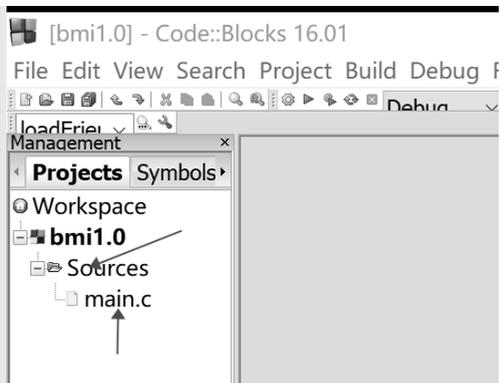


图 2.12 代码编写界面

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

完成 C 语言开发环境的安装与配置管理，其运行程序能够正确显示“Hello World!”的效果，如图 2.13 所示。

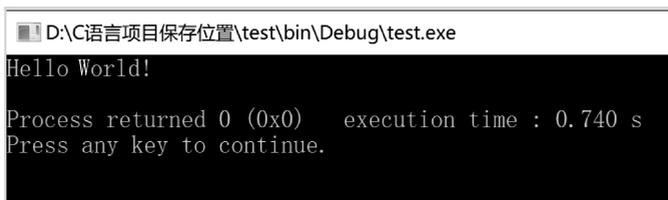


图 2.13 开发环境安装成功显示

#### ● 验收结果

序号	验收内容	实现效果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	开发环境安装与置换情况					
3	掌握知识的情况					
4	程序运行情况					
5	团队协作					

说明：在实现效果对应等级中打“√”。

#### ● 验收评价

.....

.....

验收签字.....

## 学习活动 6 总结拓展

### 任务总结与拓展

#### ● 实现效果

理解 C 语言的执行过程，并对 C 语言的基本语法规范有一定的认识，实现对 C 语言开发环境的安装与配置。

#### ● 技术层面

理解 C 语言程序执行流程，以及软件集成开发环境，掌握 C 语言的基本语法规范。

● 课程思政

通过本任务的学习，同学们可掌握 C 语言的执行过程，以及 C 语言开发环境的安装与配置。希望同学们除在技术层面完成任务外，还能对以下哲理有更进一步的思考。

“工欲善其事，必先利其器”的重要性，即只有做好充分的准备工作，才有成功的可能性。同时也希望同学们能带着一颗敬畏之心去看待事物。

● 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）

-----

-----

-----

-----



程序之路正式开始

下一回：程序中数据的保存

（编写你的第一个程序）

任务 2 程序中数据的保存



目标描述

任务描述

● 编写程序实现

编写程序实现人的体质指数（BMI）的计算程序。

● 技术层面

掌握常用的 C 语言数据类型。

掌握变量的定义与使用。

● 课程思政

生命与健康。

学习活动 1 接领任务

领任务单

● 任务确认

实现编写计算人的体质指数（BMI）程序，即完成自己的第一个 C 语言程序。

具体要求如下:

- (1) 程序能正确计算人的体质指数 (BMI);
- (2) 掌握 C 语言代码的使用规范 (变量命名及注释说明);
- (3) 程序能正确运行, 并应具有可扩展性。

- 确认签字

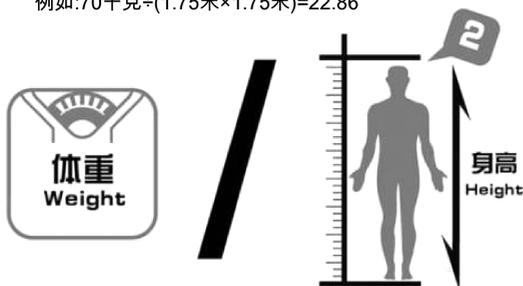
## 学习活动 2 分析任务

使用 C 语言编写一个计算人的体质指数 (BMI) 程序。

体质指数 (Body Mass Index, BMI) 是国际上常用的衡量人体胖瘦程度, 以及是否健康的一个标准。

体质指数 (BMI) = 体重 (千克) / 身高<sup>2</sup> (米)。根据体质指数对应的区间对人的健康进行指示, 如图 2.14 所示。

体质指数(BMI)=体重(千克)÷身高<sup>2</sup>(米)  
例如:70千克÷(1.75米×1.75米)=22.86



分类	BMI指数区间	健康状态
过轻	<18.5	低危险群体
正常	18.5 (含) ~ 24	正常
过重	24 (含) ~ 27	低危险群体
I 度肥胖	27 (含) ~ 30	轻度肥胖, 中危险群体
II 度肥胖	30 (含) ~ 35	中度肥胖, 重危险群体
III 度肥胖	≥35	病状肥胖

图 2.14 BMI 计算方法示意

通过分析, 实现计算人的体质指数 (BMI) 的程序, 将涉及体重数据、身高数据和计算结果这 3 个数据。

那么在程序中, 采用什么方式来保存或记录这些数据呢?

## 知识学习: C 语言的变量



### 知识学习

#### 1. 程序执行的原理

在运行时, 计算机将程序代码及程序运行的所有数据都加载到内存中, 如图 2.15 所示。

大家可以抽象地理解为内存中有很大的空间进行数据存储, 如图 2.16 所示。

这么大的空间, 怎么才能知道数据存放在哪儿呢?

程序通过使用变量的方式来对内存进行访问与数据的存储。

#### 学习笔记

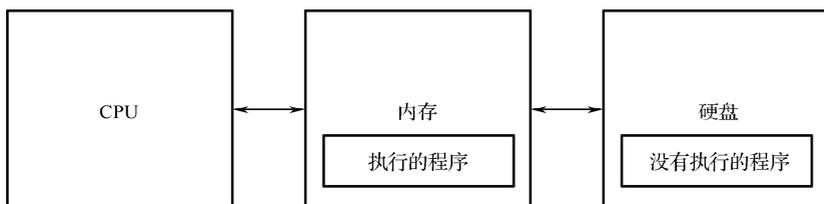


图 2.15 程序执行示意

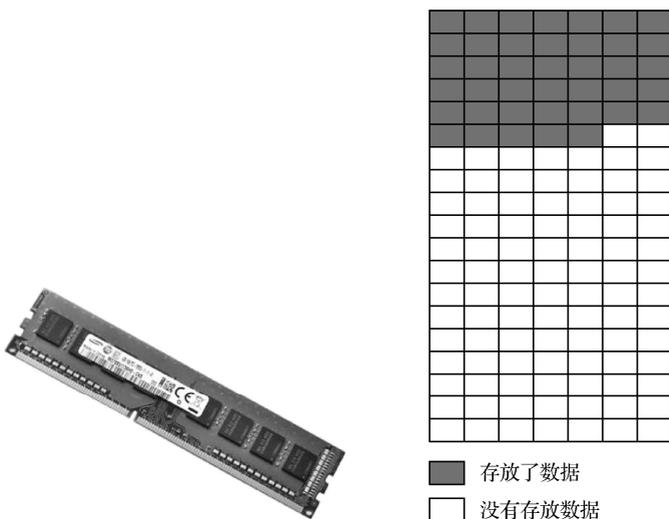


图 2.16 数据存放于内存示意

## 2. 变量

在程序中使用变量向计算机内存申请一个空间，用于存放数据，也就是说，程序中的变量对应着内存中某一个特定空间。

特定空间是由一个或多个连续字节组成的。

不同数据存入具有不同内存地址的空间，并且相互独立。

每个变量都有个名称，叫变量名。在程序中通过变量名实现对数据的写入与读取，如图 2.17 所示。

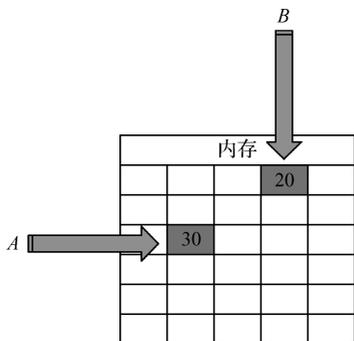


图 2.17 变量访问内存示意

### 3. 变量的命名

C 语言规定变量名 (标示符) 只能由字母、数字和下画线 3 种字符组成, 且第一个字符必须为字母或下画线。

变量名不能包含除 “\_” 以外的任何特殊字符, 如%、#等。同时也不可以使用 C 语言系统已使用的保留字。

正确的变量名: Radium、salary、AotuGun、Knotted\_Wool、H301。

非法的变量名: 9\_Ball、6\_pack、Hash!。

变量命名时, 最好做到见名知意。

### 4. 数据类型

在现实生活中数据是有类型之分的, 在 C 语言中也一样, 不同的数据可代表不同的数据类型, 也可以说数据的类型决定了数据范围。

不同类型的变量, 在内存中申请的空间大小也是不一样的, 只有类型一致的数据才会放入相应的空间中。

C 语言常用的数据类型如图 2.18 所示。

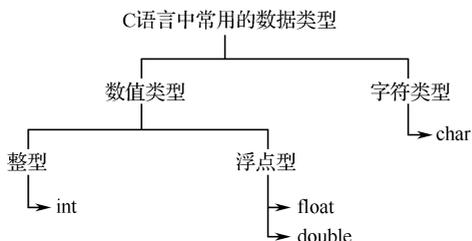


图 2.18 C 语言常用的数据类型

### 5. 变量定义

变量必须先声明, 才可以使用。声明变量有以下 3 种方式。

(1) 声明变量, 即定义变量名, 其格式:

`DataType variableName;`

如 `int stuAge;`

(2) 初始化变量, 即给变量赋初始值, 其格式:

`DataType variableName = value;`

如 `int stuAge=18;`

(3) 变量声明及初始化, 也可以这样, 其格式:

`DataType variableName;`

`variableName = value;`

如: `int stuAge;`

`stuAge=18;`

另外, 同一种类型的变量可以同时声明多个, 如:

//一次声明多个同类型变量

`int stuAge,stuWidth, stuHeight;`

`stuAge=18;`

<pre>stuWidth=65; StuHeight =175;</pre> <p><b>6. 变量的使用</b> 计算兄妹共有多少钱？</p> <pre>int gg,mm, sum; gg=1500; mm=2200; sum=gg+mm;</pre> <p>求长方形的面积。</p> <pre>float width=5.3f; float height=8.1f; float s=width*height;</pre> <p>求圆的面积。</p> <pre>double r =4.9 ; double area = 3.1415926 * r * r;</pre>	<p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p> <p>-----</p>
---	---

### 学习活动 3 制定方案

#### 实现本任务方案

● 实现思路

通过对本任务的分析及相关知识学习，制定方案如下：

- (1) 创建项目；
- (2) 定义 3 个 float 类型的变量（分别用于存放体重数据、身高数据和计算后的 BMI）；
- (3) 计算结果。

● 实现步骤

- (1) 在 CodeBlocks 软件中创建一个新项目，项目名称为 BMI。
- (2) 在 main.c 文件中，按实现思路编写代码。

### 学习活动 4 实施实现

#### 任务实现

● 实现代码

- (1) 创建项目。打开 CodeBlocks 软件，创建一个新的控制台项目，项目名称输入为 BMI。
- (2) 打开项目中的 main.c 文件，进入编辑界面。
- (3) 在 main()中按实现思路完成本任务，其代码如下：

```
int main()
{
    //定义变量
    float height,weight,bmi;
    height=1.71f; //身高 1.71 米
```

```

weight=65; //体重 65 千克
//计算 BMI
bmi=weight/(height*height);

return 0;
}
    
```

(4) 运行程序。

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

编写 C 语言程序，实现 BMI 的计算程序。

按制定的方案进行任务实现，在正确的情况下，任务实现的效果如图 2.19 所示，但没有显示出计算的结果。

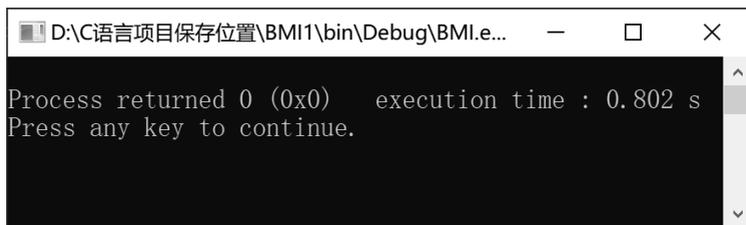


图 2.19 任务运行效果

#### ● 验收结果

序 号	验 收 内 容	实 现 效 果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	使用代码的规范性（变量命名、注释说明）					
3	掌握知识的情况					
4	程序性能及健壮性					
5	团队协作					

说明：在实现效果对应等级中打“√”。

#### ● 验收评价

验收签字.....

## 学习活动 6 总结拓展

### 任务总结与拓展

- 实现效果

本任务让同学们完成了自己的第一个 C 语言程序的编写，实现了人的体质指数（BMI）的计算，但是没有显示出结果。

- 技术层面

掌握 C 语言的变量和数据类型。

- 课程思政

程序语言可以实现无穷的功能，本次选择 BMI 这个程序，是因为健康是我们最宝贵的财富。

希望同学们不但要学会相应的 C 语言技术知识，还要明白生命与健康的意义。

- 教学拓展

本任务没有将计算结果进行显示，同学们可以试着将其实现。

- 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）

-----

-----

-----

-----



老师，什么也没有显示

下一回：程序中数据的显示

（让程序可以显示结果）

## 任务 3 程序中数据的显示



### 目标描述

#### 任务描述

- 编写程序实现

对上个任务完成的 BMI 程序进行优化，实现对计算结果的显示。

- 技术层面

掌握程序输出的方法。

掌握 printf()输出函数的方法。

- 课程思政  
细节决定成败。

## 学习活动 1 接领任务

### 领任务单

- 任务确认  
通过对上次编写的 BMI 程序进行改进，实现程序的显示。  
具体要求如下：  
(1) 实现对上次编写的 BMI 程序的显示；  
(2) 掌握 C 语言代码的使用规范（变量命名及注释说明）；  
(3) 程序能正确运行，并应具有可扩展性。
- 确认签字

## 学习活动 2 分析任务

对上次编写的 BMI 程序进行改进，实现对数据的显示功能。

上次任务实现的 BMI 程序，虽然完成了计算，但运行时什么也不显示。这是因为我们只完成了计算部分，并没有编写将结果输出显示的代码，所以没有在屏幕上显示出结果。

在程序的世界中，输出是指输出程序计算的结果，包括的内容如下：

- (1) 输出显示到屏幕上；
- (2) 输出到文件中保存；
- (3) 输出到打印机中打印。

本任务详细介绍 C 语言中的格式输出函数，实现在屏幕上显示结果。

## 知识学习：C 语言格式输出 printf()



知识学习

### 1. 格式输出 printf()

在 C 语言中使用格式输出 printf(), 实现将数据按指定的格式显示在屏幕上。

格式输出的函数语法：

**printf("格式控制", 输出项);**

格式输出函数既然是按格式要求进行显示的，那么什么是格式呢？

### 2. 格式控制符

C 语言中格式字符串的一般形式为%类型。

学习笔记

-----

-----

-----

-----

-----

类 型	说 明
c	输出单个字符
d	以十进制形式输出带符号的整数（正数不输出符号）
e	以指数形式输出单、双精度实数
E	以指数形式输出单、双精度实数
f	以小数形式输出单、双精度实数
i	有符号十进制整数（与%d相同）
o	以八进制形式输出无符号整数（不输出前缀O）
s	输出字符串
x	以十六进制形式输出无符号整数（不输出前缀OX）
X	以十六进制形式输出无符号整数（不输出前缀OX）
u	以十进制形式输出无符号整数

常用的 C 语言中格式字符串的 4 种类型如下：

类 型	说 明
c	输出单个字符
d	以十进制形式输出带符号整数（正数不输出符号）
f	以小数形式输出单、双精度实数
s	输出字符串

### 3. 格式输出函数应用的示例

输出语句	输出结果	说 明
<code>printf("Code,是一种艺术");</code>	Code,是一种艺术	原样输出
<code>printf("结果是: %f", 12.04);</code>	结果是: 12.04	对应替换
<code>printf("这个数是: %d", 10);</code>	这个数是: 10	对应替换
<code>printf("一个 %c 字符", 'a');</code>	一个 a 字符	对应替换
<code>printf("你好, %s! 你的等级为: %d, 学生叫你 %s。", "罗勇", 100, "LuoSir");</code>	你好, 罗勇! 你的等级为: 100, 学生叫你 LuoSir	对应替换

说明：

(1) 没有格式符的情况下，原样输出，如

`printf("Code,是一种艺术");`

(2) 只有一个格式符时，根据格式符的要求输出，如

`printf("结果是: %f", 12.04);` //输出一个小数

`printf("这个数是: %d", 10);` //输出一个整数

`printf("一个 %c 字符", 'a');` //输出一个字符

(3) 同时有多个格式符时，格式符从第一个开始依次对应后面的输出值，

输出值之间使用逗号分开，如

`printf("你好, %s! 你的等级为: %d, 学生叫你 %s", "罗勇", 100, "LuoSir");`

第 1 个格式符 `%s` 对应第 1 个值: 罗勇;  
 第 2 个格式符 `%d` 对应第 2 个值: 100;  
 第 3 个格式符 `%s` 对应第 3 个值: LuoSir;  
 具体对应如上表所示的显示结果。

#### 4. printf 格式输出函数实战

显示枪械信息的效果如图 2.20 所示。

```

SCAR-L(自动步枪)
-----
弹药: 5.560000 mm子弹
弹夹: 30 发
扩容: 40 发
获取: 拾取
-----
威力: 41
有效射程: 600
射程: 625
后坐力: 9000
-----
  
```

图 2.20 显示枪械信息的效果

其代码如下:

```

int main()
{
    //显示数据
    printf("SCAR-L(%s)\n","自动步枪");    printf("-----\n");
    printf("弹药: %f mm 子弹\n",5.56);
    printf("弹夹: %d 发\n",30);
    printf("扩容: %d 发\n",40);
    printf("获取: 拾取\n");
    printf("-----\n");
    printf("威力: %d\n",41);
    printf("有效射程: %d\n",600);
    printf("射程: %d\n",625);
    printf("后坐力: %d\n",9000);
    printf("-----\n");
    return 0;
}
  
```

### 学习活动 3 制定方案

#### 实现本任务方案

##### ● 实现思路

通过对本任务的分析及相关知识学习,制定方案如下:

- (1) 打开上次任务实现的 BMI 程序;
- (2) 在上次计算 BMI 值的代码后增加显示输出结果的代码;
- (3) 运行显示结果。

### ● 实现步骤

- (1) 在 CodeBlocks 软件中打开上次实现的 BMI 项目。
- (2) 在 main.c 文件中按实现思路编写代码，完成本任务。

## 学习活动 4 实施实现

### 任务实现

#### ● 实现代码

- (1) 启动 CodeBlocks 软件，打开 BMI 项目。

操作步骤如下：

执行“File”→“Open...”，如图 2.21 所示。

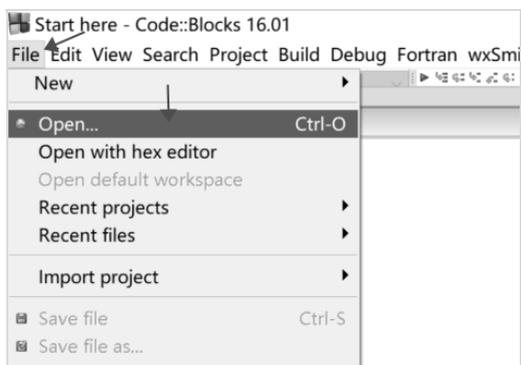


图 2.21 打开项目

找到保存 BMI 程序的位置，单击“打开”按钮，如图 2.22 所示。

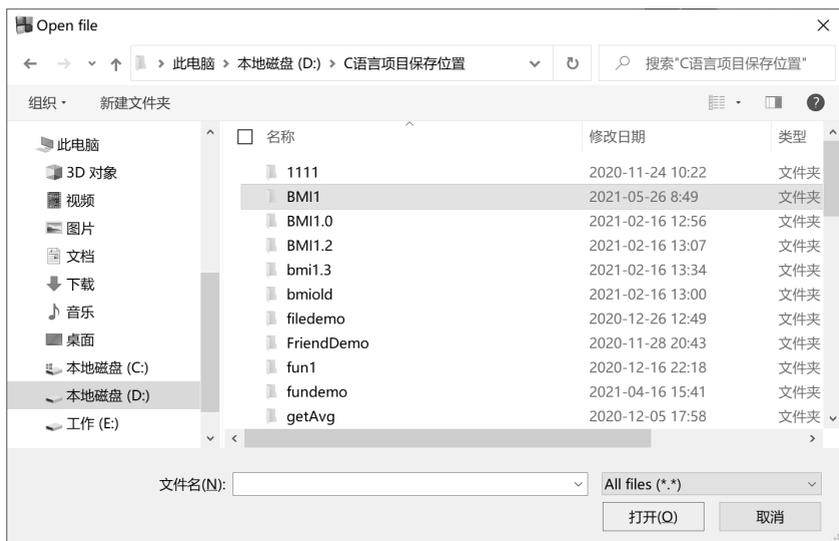


图 2.22 选择项目保存位置

选择 BMI.cbp 文件，单击“打开”按钮，如图 2.23 所示。

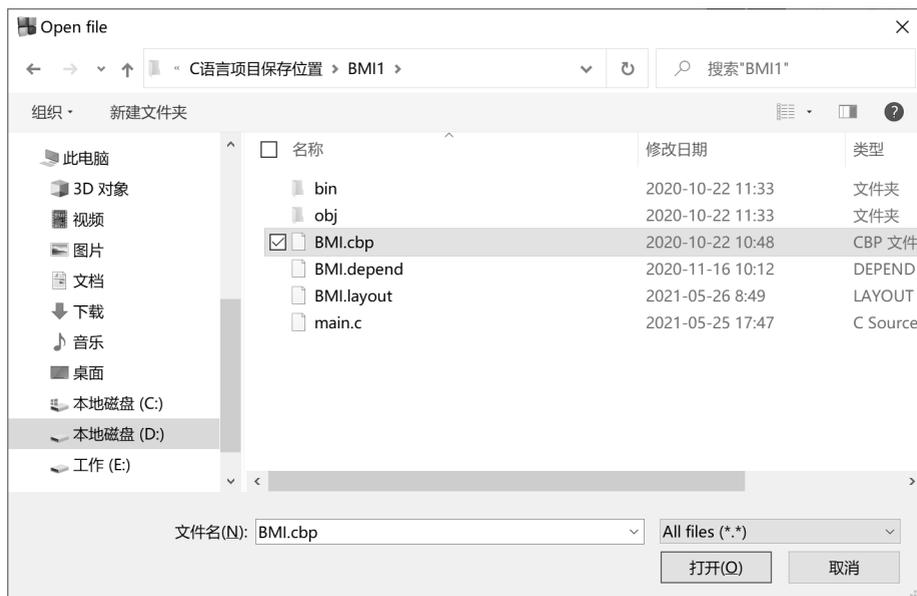


图 2.23 选择项目文件

(2) 打开项目中的 main.c 文件，进入编辑界面。

(3) 在 main() 中按实现思路完成本任务，代码如下：

```
int main()
{
    //定义变量
    float height,weight,bmi;
    height=1.71f; //身高 1.71 米
    weight=65; //体重 65 千克
    //计算 BMI 值
    bmi=weight/(height*height);

    //显示 BMI 值(本任务完成)
    printf("你的 BMI 值是: %f",bmi);
    return 0;
}
```

(4) 运行程序。

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

实现 BMI 程序显示结果的功能。

按制定方案进行任务实现，在正确的情况下，任务运行效果如图 2.24 所示，可以正确

地显示出计算结果。

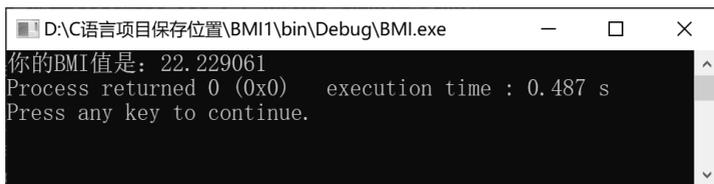


图 2.24 任务运行效果

● 验收结果

序 号	验收 内容	实 现 效 果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	使用代码的规范性（变量命名、注释说明）					
3	掌握知识的情况					
4	程序性能及健壮性					
5	团队协作					

说明：在实现效果对应等级中打“√”。

● 验收评价

.....  
 .....  
 验收签字.....

学习活动 6 总结拓展

任务总结与拓展

● 实现效果

对上次任务完成的 BMI 程序进行显示功能的实现。

● 技术层面

介绍了 C 语言的格式输出 printf() 的相关知识。通过该函数可以实现将数据显示在屏幕上，但一定要在格式规定的要求下进行。因此，同学们在使用这个函数时一定要细心。应根据显示的数据类型选择好对应的格式符，同时显示多个数据时，要有对应个数的格式符。

● 课程思政

希望同学们能将该函数用于生活的方方面面，记住细节决定成败，

● 教学拓展

完成了对计算结果的显示，但这个程序的结果是不变的。请同学们试着修改程序，让不同的数据“动”起来。

- 任务小结 (请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等)

---



---



---



### 是否发现了什么问题

目前, 这个程序运行的结果是不变的。能不能实现根据用户的输入值进行动态计算呢?

### 下一回: 程序中数据的输入

(实现用户输入数据, 让结果“动”起来)

## 任务 4 程序中数据的输入



### 目标描述

#### 任务描述

- 编写程序实现  
对上次任务完成的 BMI 程序继续进行优化, 实现程序在运行时, 由用户输入身高和体重, 从而实现不同的计算结果。
- 技术层面  
掌握程序输入的方法。  
掌握 scanf() 输入函数的方法。
- 课程思政  
善于思考。

### 学习活动 1 接领任务

#### 领任务单

- 任务确认  
对上次任务编写的 BMI 程序进行改进, 实现程序在运行时, 由用户输入身高和体重的数据后实现不同的计算结果。  
具体要求如下:  
(1) 对上次任务实现的 BMI 程序进行改进;  
(2) 在程序运行时, 由用户输入身高和体重;



<pre>float R=0,S=0; //输入数据 printf("请输入圆的半径 r:"); scanf("%f",&amp;R); //输入数据到变量中 //计算 S=3.14*R*R; //显示 printf("圆面积为:%f",S); return 0; }</pre>	<hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/>
--	---

### 学习活动 3 制定方案

#### 实现本任务方案

● 实现思路

通过对本任务的分析及相关知识学习，制定方案如下：

- (1) 打开上次任务实现的 BMI 程序；
- (2) 将身高和体重赋初值为 0；
- (3) 使用格式输入身高和体重，并保存到对应的变量中；
- (4) 运行程序。

● 实现步骤

- (1) 在 CodeBlocks 软件中打开上次实现的 BMI 项目。
- (2) 在 main.c 文件中，按实现思路编写代码。

### 学习活动 4 实施实现

#### 任务实现

● 实现代码

- (1) 启动 CodeBlocks 软件，打开 BMI 项目。
- (2) 打开项目中的 main.c 文件，进入编辑界面。
- (3) 在 main()中按实现思路完成本任务，其代码如下：

```
int main()
{
    //定义变量
    float height,weight,bmi;
    //-----本任务-----//
    height=0; //身高
    weight=0; //体重

    //输入数据
```

```
printf("请输入您的身高（单位：米）：");  
scanf("%f",&height); //输入数据到变量中  
printf("请输入您的体重（单位：千克）：");  
scanf("%f",&weight); //输入数据到变量中  
  
//-----本任务-----//  
  
//计算 BMI  
bmi=weight/(height*height);  
//显示 BMI 值  
printf("你的 BMI 值是： %f",bmi);  
return 0;  
}
```

(4) 运行程序。

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

实现人的体质指数（BMI）程序，在运行时能够动态输入数据的功能。

按制定方案进行任务实现，在正确的情况下，其效果如图 2.25 所示，用户输入不同的数据，相应的结果也不同。



图 2.25 任务运行效果

● 验收结果

序 号	验 收 内 容	实 现 效 果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	使用代码的规范性 (变量命名、注释说明)					
3	掌握知识的情况					
4	程序性能及健壮性					
5	团队协作					

说明：在实现效果对应等级中打“√”。

● 验收评价

.....  
 .....

验收签字.....

学习活动 6 总结拓展

任务总结与拓展

● 实现效果

对上次完成的 BMI 程序进行优化，实现了程序在运行时能够动态输入数据的功能。

● 技术层面

介绍了 C 语言的格式输入 scanf() 的相关知识。通过该函数可以将从键盘输入的数据保存到程序的变量中，从而实现用户输入。

● 课程思政

虽然人的体质指数程序已实现了输入 (Input)、计算/处理 (Process)、输出 (Output) 的过程，但程序计算显示出的 BMI 有什么意义呢？我们在做任何事情时，都应该从多个角度思考问题，让自己成为一个善于思考的人。

● 任务小结 (请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等)

.....  
 .....



## 让程序理解 BMI 的含义

下一回：程序中数据的计算

(认识程序中的计算与表达)

### 任务 5 程序中数据的计算



#### 目标描述

##### 任务描述

- 目标实现

通过上次任务完成的 BMI 程序引出的问题，实现程序对数据区间的表示。

- 技术层面

掌握算术运算。

掌握关系运算。

掌握逻辑运算。

掌握赋值运算。

- 课程思政

磨刀不误砍柴工。

#### 学习活动 1 接领任务

##### 领任务单

- 任务确认

实现对 C 语言运算符及表达的学习，实现体质指数区间程序的表示。

具体要求如下：

(1) 根据 BMI 参照表，利用关系运算符和逻辑运算符实现对 BMI 值区间的表示；

(2) 不要求在程序中完成，可在实现单中手写填入；

(3) 书写规范，说明清楚。

- 确认签字

#### 学习活动 2 分析任务

使用 C 语言程序来表示 BMI 值的区间，我国 BMI 指数对照表如图 2.26 所示。



## 表达与运算

我国BMI指数对照表		
分类	BMI指数区间	健康状态
过轻	<18.5	低危险群体
正常	18.5(含)~24	正常
过重	24(含)~27	低危险群体
I度肥胖	27(含)~30	轻度肥胖, 中危险群体
II度肥胖	30(含)~35	中度肥胖, 重危险群体
III度肥胖	≥35	病状肥胖

图 2.26 我国 BMI 指数对照表

## 知识学习: C 语言运算符及表达式



## 知识学习

### 1. 算术运算符

算术运算符是指处理四则运算的符号。

运算符	描述	举例 (设 A=10, B=2)
+	相加求和	A+B, 结果为 12
-	相减求差	A-B, 结果为 8
*	相乘求积	A*B, 结果为 20
/	相除求商	A/B, 结果为 5
%	求整除后的余数	A%B, 结果为 0
++	自增, 每次+1	A++, 结果为 11
--	自减, 每次-1	B--, 结果为 1

### 2. 关系运算符

可以把关系运算理解为一种“判断”, 其判断的结果是“真”或“假”。

运算符	描述	举例 (设 A=10, B=2)
==	等于。判断两个操作数是否相等, 若相等则结果为真; 若不相等则结果为假	A==B, 结果为假
!=	不等于。判断两个操作数是否相等, 若不相等则结果为真	A!=B, 结果为真
>	大于	A>B, 结果为真
<	小于	A<B, 结果为假
>=	大于或等于	A>=B, 结果为真
<=	小于或等于	A<=B, 结果为假

### 3. 逻辑运算符

逻辑运算符是指处理逻辑运算的符号。

### 学习笔记

运算符	描述	举例 (设: A=10, B=2, C=5)
&&	逻辑与。两个操作数进行与运算, 若两操作数都为真, 则结果为真	A>B && B>C, 结果为假
	逻辑或。两个操作数进行或运算, 若其中一个操作数为真, 则结果为真	A>B    B>C, 结果为真
!	逻辑非。单操作运行, 真变假, 假变真	!(A>B), 结果为假 !(B>C), 结果为真

为了更好地理解逻辑运算的含义, 以下分别对逻辑与、逻辑或、逻辑非 3 种运算进行举例 (假设, 真为 0, 假为 1)。形成 3 种运算的真值表, 如图 2.27 所示。

A	B	A&&B	A	B	A  B	A	!A
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

逻辑与

逻辑或

逻辑非

图 2.27 逻辑运算的真值表

#### 4. 表达式

由运算符组成的式子称为表达式。

##### (1) 算术表达式

如: `int a=2+6*3+15;`

结果为 `a=35`。

##### (2) 关系表达式

如:

`int x=10;`

`int y=11;`

`x>0`, 结果为真 (1)。

`y!=6+5`, 结果为假 (0)。

##### (3) 逻辑表达式

如:

`int x=10;`

`int y=11;`

`(x>0 && y>0)`, 结果为真 (1)。

##### (4) 赋值表达式

如:

`int a=65;`

##### (5) 其他

假设:

```
int a=5;
那么：
+=: a+=1; 相当于 a=a+1; 结果为 a=6;
-=: a-=1; 相当于 a=a-1; 结果为 a=4;
*=: a*=2; 相当于 a=a*2; 结果为 a=10;
/=: a/=1; 相当于 a=a/1; 结果为 a=5。
```

### 学习活动 3 制定方案

#### 实现本任务方案

##### ● 实现思路

通过对本任务的分析及相关知识学习，制定方案如下：

- (1) 假设体质指数值保存在 BMI 变量中；
- (2) 使用关系运算符和逻辑运算符实现 BMI 值区间的表示。

### 学习活动 4 实施实现

#### 任务实现

##### ● 实现代码

假设体质指数保存在 BMI 变量中，使用关系运算符和逻辑运算符完成 BMI 值区间的表示，在以下表格中完成填写工作。

分 类	BMI 指数区间	程序中对应的表达式（假设以变量 BMI 来表示结果）
过轻	<18.5	
正常	18.5（含）~24	
过重	24（含）~27	
I 度肥胖	27（含）~30	
II 度肥胖	30（含）~35	
III 度肥胖	≥35	

在以上表格中完成填写。

### 学习活动 5 测试验收

#### 任务测试验收单

##### ● 实现效果

本任务不要求在程序中完成。实现对 BMI 值区间使用程序的方式来表示，实现后的参考如下所示。

分 类	BMI 指数区间	程序中对应的表达式（假设以变量 BMI 来表示结果）
过轻	<18.5	BMI < 18.5
正常	18.5（含）~24	BMI >= 18.5 && BMI < 24
过重	24（含）~27	BMI >= 24 && BMI < 27
I 度肥胖	27（含）~30	BMI >= 27 && BMI < 30
II 度肥胖	30（含）~35	BMI >= 30 && BMI < 35
III 度肥胖	≥35	BMI >= 35

● 验收结果

序 号	验 收 内 容	实 现 效 果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	使用代码的规范性（变量命名、注释说明）					
3	掌握知识的情况					
4	程序性能及健壮性					
5	团队协作					

说明：在实现效果对应等级中打“√”。

● 验收评价

验收签字.....

## 学习活动 6 总结拓展

### 任务总结与拓展

● 实现效果

使用 C 语言的关系运算符和逻辑运算符实现对 BMI 值区间的表示。

● 技术层面

掌握 C 语言中常用运算符及其运算表达式，在这个实现过程中，始终以 BMI 程序引出所学内容。

● 课程思政

希望同学们在处理任何事务时都不急不躁，确定目标后一步一步地去实现或解决，做个静心之人。另外，希望同学们明白“磨刀不误砍柴工”的道理，打牢基础，为后续的学习奠定基础。

● 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）

---

---

---

---

---

---

---

---



### 让程序能够做出判断

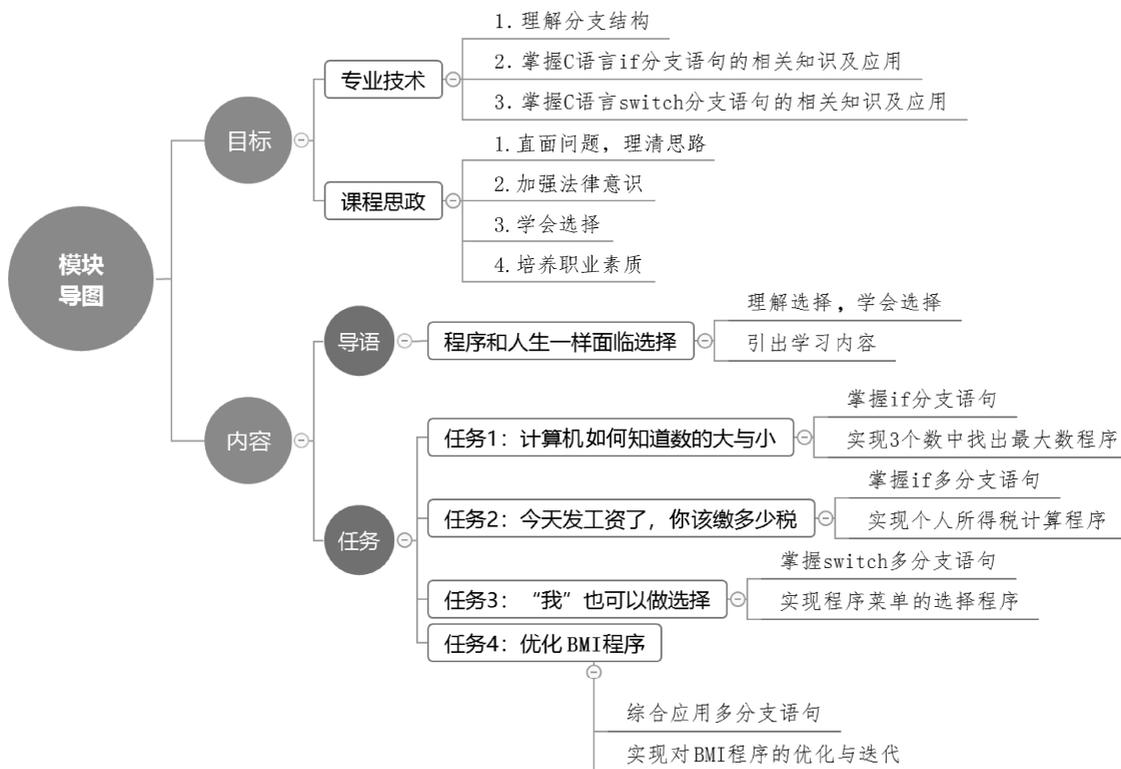
实现区间的表示后，程序怎么才能知道属于哪个区间呢？

**下一回：**进入模块 3 C 语言程序中的分支结构应用

（开启程序的“选择”模式，让程序能够进行判断）

# 模块 3

## C 语言程序中的分支结构应用



## 项目导语：程序和人生一样面临选择



### 1. 生活中的选择

在日常生活中，我们经常需要进行各种选择，人生就是一个不断选择的过程。

选择就意味着不得不放弃一些东西，同时也是人生的价值所在。因为我们可以不断选择中茁壮成长，规划自己的人生，走一条适合自己的路，如图 3.1 所示。



#### 人生之路

人生之路千万条，  
我的人生该如何设计呢？



图 3.1 人生之路

在生活中，我们所见到的路标也是一种选择，如图 3.2 所示。



#### 我去哪

它只是一个路标吗？



图 3.2 生活中的路标

路标指示着能到达的目的地，但我们不可能同时选择去不同的地方，只能有一个目的地，那个决定我们做出选择的因素就是动机（条件）。例如，我要去教学楼听课，所以选择去教学楼的路；我要回宿舍休息，那就选择去学生宿舍的路。

所以，我们内心的动机决定着如何去选择。

### 2. 程序中的选择

上面给大家分享了一些生活中的选择，接下来，说说程序中的选择。

程序中的结构分为顺序结构、分支结构和循环结构。下面介绍其中的一种结构，即分支结构（或选择结构）。

分支结构说明如下：

- ① 如果条件成立，则选择执行程序块 A 代码，然后再执行分支结构外的程序块 C；
- ② 如果条件不成立，则选择执行程序块 B 代码，然后再执行分支结构外的程序块 C；

③ 如图 3.3 所示，程序块 A 和程序块 B 构成了一个选择，只能选择一个来执行，决定选择的条件就是条件（动机）。

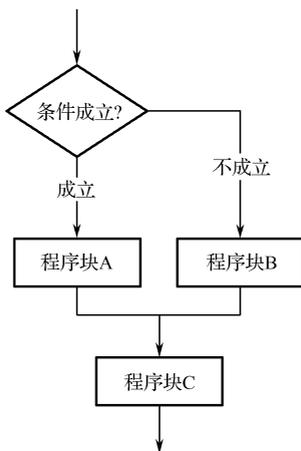


图 3.3 分支结构流程



### 实现选择的语句

为了全面介绍程序中的选择知识，本模块共设计以下 4 个任务。

任务 1：计算机如何知道数的大与小。

任务 2：今天发工资了，你该缴多少税。

任务 3：“我”也可以做选择。

任务 4：优化 BMI 程序。

## 任务 1 计算机如何知道数的大与小



### 目标描述

#### 任务描述

- 编写程序实现

输入：3 个任意整数。

判断：找出 3 个数中的最大数，如图 3.4 所示。

输出：在屏幕中显示这个最大数。

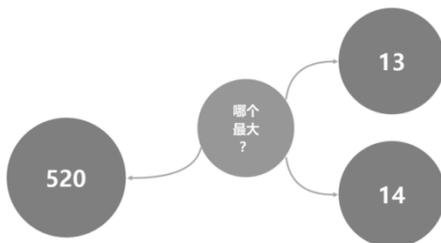


图 3.4 找出 3 个数中的最大数

- 技术层面  
掌握 if 分支语句的意义及应用。
- 课程思政  
直面问题，理清思路。

## 学习活动 1 接领任务

### 领任务单

- 任务确认  
编写 C 语言程序，实现任意输入 3 个整数，并能找出最大数。  
具体要求如下：  
(1) 程序最终能正确找出并显示，任意输入 3 个整数中的最大数；  
(2) 掌握 C 语言代码的使用规范（变量命名及注释说明）；  
(3) 程序能正确运行，并应具有可扩展性。
- 确认签字

## 学习活动 2 分析任务

编写 C 语言程序，实现任意输入 3 个整数，让程序判断出这 3 个数中的最大数并显示，具体分析如下：

- (1) 输入 3 个数，并保存到对应的变量中；
- (2) 判断找出最大数；
- (3) 显示最大数，程序结束。

## 知识学习: if 分支语句



if 分支语句是 C 语言实现分支的语句之一，在给定的条件成立时，选择执行某段程序。

### 1. if 语句实现的单分支结构

if(条件表达式)

```
{
    语句块 1
}
...
```

说明：

- (1) if 是命令动词；
- (2) ()里写的是条件表达式；
- (3) {}限定选择语句的范围；

学习笔记

---

---

---

---

---

---

---

---

---

---



### 学习活动 3 制定方案

#### 实现本任务方案

##### ● 实现思路

通过对本任务的分析及相关知识的学习,制定方案如下:

- (1) 定义 4 个整型变量, 并进行初始化;
- (2) 通过 scanf() 输入 3 个任意数, 并保存到对应的变量中;
- (3) 利用 if 语句实现数的大小判断处理;

思路: 比较第 1 个数和第 2 个数找到两者的较大值, 然后将此较大值与第 3 个数比较, 最终确定 3 个数中的最大数。

- (4) 输出这 3 个数中的最大数。

##### ● 实现步骤

- (1) 在 CodeBlocks 软件中创建一个新项目, 项目名称为 getMax。
- (2) 在 main() 中按实现思路编写代码。

### 学习活动 4 实施实现

#### 任务实现

##### ● 实现代码

- (1) 打开 CodeBlocks 软件, 创建一个新的控制台项目, 项目名称输入为 getMax。
- (2) 打开项目中的 main.c 文件, 进入编辑界面。
- (3) 在 main() 中按实现思路完成任务, 其代码如下:

```
int main()
{
//1. 定义 4 个变量
int numone,numtwo,numthree, w;
//2. 输入 3 个数
printf("请输入 3 个任意数\n");
scanf("%d",&numone); //输入第 1 个数
scanf("%d",&numtwo); //输入第 2 个数
scanf("%d",&numthree); //输入第 3 个数
//3. 将第 1 个数和第 2 个数进行比较, 找出最大数并保存到 w 变量中
if(numone<numtwo) {
    w=numtwo;
} else
{
    w=numone;
}
//第 1 个数和第 2 个数中的最大数与第 3 个数进行比较, 找出最大数
if(w<numthree){
    printf("这 3 个数中, 最大的数是%d",numthree);
```

```

    }else
    {
        printf("这3个数中,最大的数是%d",w);
    }
    return 0;
}

```

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

编写 C 语言程序,实现找到任意输入 3 个整数的最大数并输出。

按制定方案进行任务实现,在正确的情况下,任务实现的效果如图 3.5 所示。

```

请输入3个任意数
13
520
14
这3个数中,最大的数是:520Press any key to continue

```

图 3.5 输出任意 3 个数中最大数程序实现的效果

#### ● 验收结果

序号	验收内容	实现效果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	开发环境安装与置换情况					
3	掌握知识的情况					
4	程序运行情况					
5	团队协作					

说明:在实现效果对应等级中打“√”。

#### ● 验收评价

.....

.....

验收签字.....

## 学习活动 6 总结拓展

### 任务总结与拓展

- 实现效果

编写 C 语言程序，实现找到任意输入 3 个整数的最大数并输出。

- 技术层面

详细介绍了 if 语句的单分支、双分支结构的实现。

单分支：if 分支语句的条件成立，执行分支语句内的内容，否则跳过不执行。

双分支：if...else...结构，如果条件成立则执行 if 后面的语句；如果条件不成立则执行 else 后面的语句，二选一。

- 课程思政

在程序中的每次任务都是先提出问题，然后通过分析，最后实现任务。其实在日常生活中，我们也会经常遇到各种各样的问题，当有问题时，应面对问题，积极思考，初步形成解决问题的大致想法，然后设计出具体解决问题的思路 and 实现步骤，以最终解决问题。

- 教学拓展

通过本任务的学习，同学们掌握了 if 语句的单分支、双分支结构的使用，试着对本任务的功能进行扩展，实现将任意输入的 3 个数排序后，按从小到大的顺序输出。

- 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）

-----

-----

-----



### 要解决的问题不止两个分支时，该怎么办

下一回：你该缴多少税

（解决多分支的问题）

## 任务 2 今天发工资了，你该缴多少税



### 目标描述

#### 任务描述

- 编写程序实现

输入：税前工资总额和扣除的五险一金数额。

处理：实现个人所得税计算。

输出：显示所缴纳个人所得税的金额。



图 3.6 个人所得税

- 技术层面  
掌握 if 多分支的实现及应用。
- 课程思政  
加强法律意识。

### 学习活动 1 接领任务

#### 领任务单

- 任务确认  
编写 C 语言程序，实现个人所得税的计算。  
具体要求如下：
  - (1) 程序最终能正确展示个人应缴纳的个人所得税；
  - (2) 掌握 C 语言代码的使用规范（变量命名及注释说明）；
  - (3) 程序能正确运行，并具有可扩展性。
- 确认签字

### 学习活动 2 分析任务

编写 C 语言程序，实现个人所得税的计算。实现本任务要先了解个人所得税相关的知识。

#### (1) 个人所得税

1993 年 10 月 31 日，发布了新修改的《中华人民共和国个人所得税法》，个人所得税是国家对本国公民、居住在本国境内个人的所得和境外个人来源于本国的所得征收的一种所得税。2018 年 8 月 31 日，个人所得税起征标准调至每月 5000 元，2018 年 10 月 1 日起实施最新起征点和税率，自 2019 年 1 月 1 日起施行。

## (2) 税率

《中华人民共和国个人所得税法》规定个人综合所得适用于“个人所得税税率表一（综合所得适用）”，具体如图 3.7 所示。

级数	全年应纳税所得额	税率（%）	速算扣除数
1	不超过36000元（含）的部分	3	0
2	超过36000元至144000元（含）的部分	10	2520元
3	超过144000元至300000元（含）的部分	20	16920元
4	超过300000元至420000元（含）的部分	25	31920元
5	超过420000元至660000元（含）的部分	30	52920元
6	超过660000元至960000元（含）的部分	35	85920元
7	超过960000元的部分	45	181920元

说明：本表所称全年应纳税所得额是指依照本法第六条的规定，居民个人取得综合所得以每一纳税年度收入额减除费用60000元，以及专项扣除、专项附加扣除和依法确定的其他扣除后的余额。

图 3.7 个人所得税税率表一（综合所得适用）

## (3) 个人所得税计算的方法

应纳税所得额=月收入-5000元（起征点）-专项扣除（五险一金等）-专项附加扣除-依法确定的其他扣除

个人所得税=应纳税所得额×税率-速算扣除数

说明：新个税法规定，自2018年10月1日至2018年12月31日，纳税人的工资、薪金所得，先行以每月收入额减除费用5000元，以及专项扣除和依法确定的其他扣除后的余额为应纳税所得额，依照个人所得税税率表一（综合所得适用）按月换算后计算缴纳税款，并不再扣除附加减除费用。

因此，根据计算方法说明，将个人所得税税率表一折算为月份对应数据所示如下。

级数	应纳税所得额(元)	税率	速算扣除数(元)
1	≤3000	3%	0
2	3000~12000(含)	10%	210
3	12000~25000(含)	20%	1410
4	25000~35000(含)	25%	2660
5	35000~55000(含)	30%	4410
6	55000~80000(含)	35%	7160
7	>80000	45%	15160

例如，小李在北京上班，月收入为10000元，“五险一金”专项扣除为2000元。（简化示例，暂时不考虑2019年1月1日执行的新政策）

应纳税所得额=10000-5000-2000=3000（元）

个人所得税=3000×3%-0（速算扣除数）= 90（元）

所以，小李应该缴纳个人所得税为 90 元。



## 程序如何实现多个分支

### 知识学习：if 多分支语句



### 知识学习

if 多分支语句是 C 语言实现分支的语句之一，在给定的条件成立时，选择执行某段程序。

#### 1. if 多分支语句的语法

```
if(条件 1)
{
    语句块 1
}else if (条件 2)
{
    语句块 2
}else if (条件 3)
{
    语句块 3
}else if (条件 4)
{
    语句块 4
}
...

```

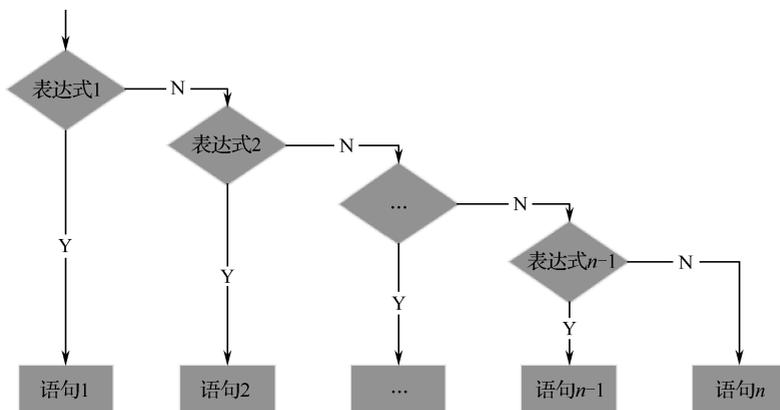


图 3.8 if 多分支语句流程

#### 说明：

多分支指的是哪个条件成立，就执行哪个语句，即实现多选一执行，如下所示：

如果执行语句 1：条件 1 成立时

执行语句 2：条件 1 不成立，条件 2 成立时

...以此类推

#### 学习笔记

## 2. if 多分支语句举例

```
int main()
{
    int score;
    printf("请输入你的分数:");
    scanf("%d",&score);
    if(score>=90)
    {
        printf("你的等级是: A");
    }else if(score>=80)
    {
        printf("你的等级是: B");
    }else if(score>=70)
    {
        printf("你的等级是: C");
    }else if(score>=60)
    {
        printf("你的等级是: D");
    }else
    {
        printf("你的等级是: E");
    }
    return 0;
}
```

### 说明:

此例实现输入百分制分数，对应显示其等级。

将输入的分数保存到 score 变量中。

如果 score=95，则第 1 个条件成立，显示等级为 A；

如果 score=85，则第 1 个条件不成立，第 2 个条件成立，显示等级为 B；

如果 score=71，则第 1、第 2 个条件不成立，第 3 个条件成立，显示等级为 C；

如果 score=64，则第 1、第 2、第 3 个条件不成立，第 4 个条件成立，显示等级为 D；

如果 score=58，则第 1、第 2、第 3、第 4 个条件不成立，执行最后 1 个 else 后的语句，显示等级为 E。

## 学习活动 3 制定方案

### 实现本任务方案

#### ● 实现思路

##### 1. 确定个税公式

应纳税所得额=工资总额-起征额（5000）-专项扣除（五险一金等）

个人所得税=应纳税所得额×适用税率-速算扣除数

## 2. 程序实现思路

(1) 定义程序使用到的变量，并进行初始化（4个）。

它们分别用于保存工资、五险一金、应纳税所得额和个人所得税。

(2) 屏幕提示输入工资、五险一金扣除金额，可获取数据并保存到对应的变量中。

(3) 计算应纳税所得额。

应纳税所得额=工资总额-起征额（5000）-专项扣除（五险一金等）

(4) 利用 if 语句多分支实现计算个人所得税。

(5) 显示应缴个人所得税金额，程序结束。

### ● 实现步骤

(1) 在 CodeBlocks 中创建一个新项目，项目名称为 IncomeTax。

(2) 在 main()中按实现思路编写代码。

## 学习活动 4 实施实现

### 任务实现

#### ● 实现代码

(1) 打开 CodeBlocks 软件，创建一个新的控制台项目，项目名称输入为 IncomeTax。

(2) 打开项目中的 main.c 文件，进入编辑界面。

(3) 在 main()中按实现思路完成任务，代码如下。

```
int main()
{
    //1. 定义程序使用到的变量
    float dSalary=0; //工资
    float dDeduction=0; //五险一金扣除金额
    float dY=0; //应纳税所得额
    float dTax=0; //应交个税

    //2. 输入工资和扣除项
    printf("请输入您本月工资总金额:");
    scanf("%f",&dSalary);
    printf("请输入五险一金扣除金额:");
    scanf("%f",&dDeduction);
    //3. 计算应纳税所得额
    //（实际收入-纳税基数-五险一金扣除金额）
    dY=dSalary-5000-dDeduction;
    //4. if 多分支实现个税计算
    if(dY<=0)
    {
        dTax=0;
    }else if(dY<=3000)
```

```
{
    //应纳税所得额*税率-速算扣除数
    dTax=dY*0.03-0;
} else if(dY>3000 && dY<=12000)
{
    dTax=dY*0.1-210;
} else if(dY>12000 && dY<=25000)
{
    dTax=dY*0.2-1410;
} else if(dY>25000 && dY<=35000)
{
    dTax=dY*0.25-2660;
} else if(dY>35000 && dY<=55000)
{
    dTax=dY*0.3-4410;
} else if(dY>55000 && dY<=80000)
{
    dTax=dY*0.35-7160;
} else
{
    dTax=dY*0.45-15160;
}
//5. 显示结果
printf("您本月工资: %.2f 元\n",dSalary);
printf("应缴个人所得税: %.2f 元。",dTax);
return 0;
}
```

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

利用个人所得税的计算方法,根据税前工资及相关扣除项计算出应纳税所得额。利用 if 多分支语句实现对应税率的计算,并输出个人应缴纳的个人所得税。

按制定方案进行任务实现,正确的情况下,任务实现的效果如图 3.9 所示。

```
请输入您本月工资总金额:10000元
请输入五险一金扣除金额:2000元
您本月工资: 10000.00元, 应缴个人所得税: 90.00元。
```

图 3.9 个人所得税程序实现

● 验收结果

序 号	验收内容	实现效果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	使用代码的规范性（变量命名、注释说明）					
3	掌握知识的情况					
4	程序性能及健壮性					
5	团队协作					

说明：在实现效果对应等级中打“√”。

● 验收评价

-----

-----

验收签字\_\_\_\_\_

学习活动 6 总结拓展

任务总结与拓展

● 实现效果

利用个人所得税的计算方法，根据税前工资及相关扣除项计算出应纳税所得额。利用 if 多分支语句实现了对应税率的计算，并输出个人应缴纳的个人所得税。

● 技术层面

详细介绍了 if 语句的多分支结构的实现。

if...else if...多分支结构：指哪个条件成立，则选择执行该分支下的语句，从而实现多分支的过程。

注意：

- (1) {}应配对好。
- (2) else if 之间有空格，条件后不能加分号。

● 课程思政

编写 C 语言程序让同学们实现了个人所得税的计算。

通过本任务的学习，知道了交纳个人所得税是法律赋予每个公民的权利及义务，我们要做一个不偷税、漏税的守法公民。

同时，同学们也应该认识到知情权的重要性。当自己成为一名职员时，是有查看个人收入明细知情权的。

● 教学拓展

同学们已掌握了 if 多分支结构的应用，试着举一反三完成如下任务：

超市的某些商品正在打折促销，购买这些商品可根据购买数量 ( $n$ ) 给予不同折扣，折扣

信息如图 3.10 所示。请编写程序，根据用户输入的商品数量及单价，输出用户应付的金额。

数 量	折 扣
$n < 5$	0
$5 \leq n < 10$	1%
$10 \leq n < 20$	2%
$20 \leq n < 50$	3%
$n \geq 50$	5%

图 3.10 商品数量折扣对照表

- 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）

---



---



---



---



除 if 语句外，还有其他的分支语句吗

下一回：“我”也可以做选择

（switch 多分支语句）

### 任务 3 “我”也可以做选择



#### 目标描述

#### 任务描述

- 编写程序实现  
某程序主界面菜单的选择功能。  
输入：菜单编号。  
处理：根据输入的编号进行判断。  
输出：显示对应菜单的功能。
- 技术层面  
掌握 switch 多分支语句实现及应用。
- 课程思政  
学会选择。

## 学习活动 1 接领任务

### 领任务单

#### ● 任务确认

编写 C 语言程序，实现某程序主界面的菜单操作，根据用户输入对应菜单的编号实现相应的操作。

具体要求如下：

- (1) 程序最终能正确展示程序主界面，并可根据输入菜单项编号实现相应操作；
- (2) 掌握 C 语言代码的使用规范（变量命名及注释说明）；
- (3) 程序能正确运行，并具有可扩展性。

#### ● 确认签字

## 学习活动 2 分析任务

编写 C 语言程序，实现某程序主界面的菜单操作，可根据用户输入对应菜单的编号实现相应的操作。

如输入 1：进行添加操作；

如输入 2：进行修改操作；

如输入 3：进行删除操作；

如输入 0：退出系统。

这里有 4 种可能性，也就是 4 个分支，虽然使用 if 多分支语句可以实现，但今天我们使用 C 语言中的另外一个多分支语句 switch 来实现。

## 知识学习：switch 多分支语句



### 知识学习

利用 if 语句可以实现多分支的结构，但如果分支较多时，则嵌套的 if 语句层就会较多，令人理解起来较困难，所以 C 语言还提供了一个 switch 多分支语句，专门用于处理多分支结构的程序设计。

#### 1. switch 的语法

```
switch(表达式)
{
    case 常量表达式 1: {语句 1} break;
    case 常量表达式 2: {语句 2} break;
    case 常量表达式 3: {语句 3} break;
    case 常量表达式 4: {语句 4} break;
    ...
    case 常量表达式 n: {语句 n} break;
    default: {语句 n+1} break;
```

#### 学习笔记

---

---

---

---

---

---

---

---

---

---

```
}  
...
```

**说明:**

- (1) 用 `switch` 计算表达式的值;
- (2) 将这个值与 `case` 的常量表达式的值进行比较;
- (3) 如果这个值与某个 `case` 后的值相同, 则执行对应 `case` 后的语句, 遇到 `break` 时结束;
- (4) 如果这个值与所有 `case` 后的值都不符合, 则执行 `default` 后的语句。

**2. switch 多分支语句举例**

```
int main()  
{  
    int score;  
    printf("请输入你的分数:");  
    scanf("%d",&score);  
  
    switch(score/10)  
    {  
        case 10:  
        case 9:  
        {  
            printf("成绩等级为: A");  
        }  
        break;  
        case 8:  
        {  
            printf("成绩等级为: B");  
        }  
        break;  
        case 7:  
        {  
            printf("成绩等级为: C");  
        }  
        break;  
        case 6:  
        {  
            printf("成绩等级为: D");  
        }  
        break;  
        default:  
        {  
            printf("成绩等级为: E");  
        }  
        break;  
    }  
    return 0;  
}
```

**说明：**

以上示例就是根据输入的成绩分数，输出对应的 A、B、C、D、E 等级。

下面做详细分析：

(1) 将输入成绩保存到 score 变量中。

(2) 多分支 switch 语句判断 score/10 的值与哪个 case 后的值相同，就选择那个分支执行，显示分数对应的等级。

如果 score= (90~100 中的某个数)，score/10=9 或 10，则满足 case 10: 和 case 9:，结果显示等级为 A;

如果 score= (80~89 中的某个数)，score/10=8，则满足 case 8:，结果显示等级为 B;

如果 score= (70~79 中的某个数)，score/10=7，则满足 case 7:，结果显示等级为 C;

如果 score= (60~69 中的某个数)，score/10=6，则满足 case 6:，结果显示等级为 D;

如果 score= (0~59 中的某个数)，score/10R 的结果可能是(0,1,2,3,4,5)，则不满足 case 后的所有值，执行 default:，结果显示等级为 E。

(3) 总结。

switch 多分支语句中表达式的值和 case 后的值相当于进行“=”的判断，如果相等则执行该 case 分支;

每个 case 分支必须以 break 结束，否则将会依次执行后面的所有 case 中的语句，直到 break 结束，如上例中的 case 10 后就没有 break，那它将一直往下执行。

### 学习活动 3 制定方案

#### 实现本任务方案

● 实现思路

- (1) 利用 printf()在屏幕上显示对应的菜单;
- (2) 提示用户输入菜单编号;
- (3) 利用 switch 多分支语句判断输入的编号，实现程序分支。

● 实现步骤

- (1) 在 CodeBlocks 软件中创建一个新项目，项目名称为 Menu。
- (2) 在 main()中按实现思路编写代码。

## 学习活动 4 实施实现

## 任务实现

## ● 实现代码

- (1) 打开 CodeBlocks 软件，创建一个新的控制台项目，项目名称输入为 Menu。
- (2) 打开项目中的 main.c 文件，进入编辑界面。
- (3) 在 main() 中按实现思路完成任务，其代码如下。

```
int main()
{
    printf("菜单选择: \n");
    printf("1. 添加数据\n");
    printf("2. 修改数据\n");
    printf("3. 删除数据\n");
    printf("0. 退出系统\n");

    int selectno;

    printf("请输入菜单编号: ");
    scanf("%d",&selectno);

    //分支结构用于判断输入的编号
    switch(selectno)
    {
        case 1:
        {
            printf("你选择的是完成添加数据\n");
        }
        break;

        case 2:
        {
            printf("你选择的是完成修改数据\n");
        }
        break;

        case 3:
        {
            printf("你选择的是完成删除数据\n");
        }
        break;

        case 0:
        {
            printf("你选择的是退出系统\n");
        }
    }
}
```

```

break;

default :
{
    printf("你输入的编号不在菜单范围内\n");
}
break;
}
return 0;
}
    
```

### 学习活动 5 测试验收

#### 任务测试验收单

##### ● 实现效果

利用 C 语言的 switch 多分支语句，实现某程序的主界面菜单操作，根据用户输入对应菜单的编号实现相应的操作。例如，输入 1 为添加操作；输入 0 为退出系统，以实现菜单操作的多分支判断。

按制定方案进行任务实现，在正确的情况下，任务实现的界面效果如图 3.11 所示。当输入菜单编号时，系统会出现如图 3.12 所示的相应提示。

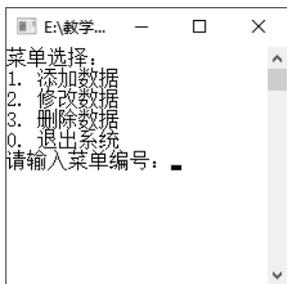


图 3.11 菜单界面效果

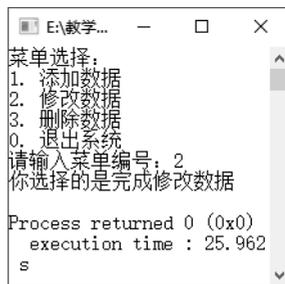


图 3.12 菜单选择效果

##### ● 验收结果

序号	验收内容	实现效果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	使用代码的规范性（变量命名、注释说明）					
		A	B	C	D	E
3	掌握知识的情况					
4	程序性能及健壮性					
5	团队协作					

说明：在实现效果对应等级中打“√”。

- 验收评价

验收签字.....

## 学习活动 6 总结拓展

### 任务总结与拓展

- 实现效果

利用 C 语言的 switch 多分支语句，实现某程序主界面的菜单操作，根据用户输入对应菜单的编号实现相应的操作。例如，输入 1 为添加操作；输入 0 为退出系统，实现菜单操作的多分支判断。

- 技术层面

介绍了 switch 多分支结构的实现。

switch...case...多分支结构：指满足哪个 case，就选择执行该分支下的语句，从而实现多分支的过程。

**注意：**

(1) switch 后面的()中是通过计算得到的一个具体的值；

(2) case 后是一个具体的值，而不能写成像 if 分支语句一样的关系表达式；

(3) switch 中表达式的值和 case 后的值相当于进行“=”的判断，如果等于则执行该 case 分支；

(4) 每个 case 分支必须以 break 结束，否则将执行 case 后的所有值，直到 break 结束，如上例中的 case 10 后就没有 break，那么它将一直往下执行。

- 课程思政

通过本任务的学习，同学们掌握了 if 和 switch 两个分支语句的相关知识。

如果程序的判断条件是一个使用关系表达式时，则应该选择 if 语句实现比较好。

如果程序的判断是一个具体明确的值时，则应该选择 switch 语句实现比较好。

在日常生活中，同学们也会面临许多选择，所以要学会选择出最适合自己的，又或当同学们遇到问题时可以多给自己设置些条件，以便能做出最佳的选择。

- 教学拓展

通过本任务的学习与实现，同学们试着完成“吃鸡游戏中枪械信息”管理界面，设置 1 为添加枪械，2 为修改枪械，3 为删除枪械，4 为查询枪械，0 为退出系统。

- 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）



## 分支结构到此结束了

### 下一回：优化体质指数（BMI）程序

（让 BMI 程序实现区间判断，并给出结论与建议）

## 任务 4 优化体质指数（BMI）程序



### 目标描述

#### 任务描述

- 编写程序实现

对前面任务实现的 BMI 程序进行优化。

根据计算所得的 BMI 值进行区间判断。

结论：属于哪种类型。

建议：给出你的建议。

- 技术层面

if 多分支语句的综合应用。

- 课程思政

职业素质。

### 学习活动 1 接领任务

#### 领任务单

- 任务确认

将所学的 if 多分支语句的知识进行综合应用，对前面任务实现的 BMI 程序进行优化。

具体要求如下：

（1）根据计算所得的 BMI 值进行区间判断，并给出结论和建议；

（2）掌握 C 语言代码的使用规范（变量命名及注释说明）；

（3）程序能正确运行，并具有可扩展性。

- 确认签字

### 学习活动 2 分析任务

将所学的 if 多分支语句知识进行综合应用，对前面任务实现的 BMI 程序进行优化。

根据计算所得的 BMI 值进行区间判断（见图 2.26）。

- (1) 结论: 属于哪种类型。
- (2) 建议: 给出你的建议。

我国 BMI 的分类有 6 种, 通过下面这些数字区间, 可以为 if 多分支结构判断条件区间, 如图 3.13 所示。



### 我国 BMI 的 6 种分类

分类	BMI 指数区间	健康状态	程序中对应的表达式 (假设以变量 BMI 来表示结果)
过轻	<18.5	低危险群体	BMI < 18.5
正常	18.5(含)~24	正常	BMI >= 18.5 && BMI < 24
过重	24(含)~27	低危险群体	BMI >= 24 && BMI < 27
I 度肥胖	27(含)~30	轻度肥胖, 中危险群体	BMI >= 27 && BMI < 30
II 度肥胖	30(含)~35	中度肥胖, 重危险群体	BMI >= 30 && BMI < 35
III 度肥胖	≥35	病状肥胖	BMI >= 35

图 3.13 BMI 指数对照表对应的条件区间

对应的 if 表示的条件区间已设置好了, 我们来为各区间进行对应的结论与建议, 如图 3.14 所示。

分类	BMI 指数区间	健康状态	结论与建议
过轻	<18.5	低危险群体	结论: 你的 BMI < 18.5, 体重过轻; 属于低危险群体 建议: 平时多注意营养, 增加体重!
正常	18.5(含)~24	正常	结论: 你的 18.5 ≤ BMI < 24, 体重正常; 属于无风险群体 建议: 别骄傲, 要保持哦!
过重	24(含)~27	低危险群体	结论: 你的 24 ≤ BMI < 27, 体重过重; 属于低危险群体 建议: 该减重啦! 加强运动。
I 度肥胖	27(含)~30	轻度肥胖, 中危险群体	结论: 你的 27 ≤ BMI < 30, 轻度肥胖; 属于中危险群体 建议: 控制饮食, 加强运动, 必须减重了。
II 度肥胖	30(含)~35	中度肥胖, 重危险群体	结论: 你的 30 ≤ BMI < 35, 中度肥胖; 属于重危险群体 建议: 减重! 减重! 减重!
III 度肥胖	≥35	病状肥胖	结论: 你的 BMI ≥ 35, 病状肥胖; 属于非常危险群体 建议: 在医生的指导下进行减重吧, 我不能给你建议了!

图 3.14 根据 BMI 指数区间得到的结论与建议



### 对于结论与建议的文字, 可以设置得更加个性

## 学习活动 3 制定方案

### 实现本任务方案

#### ● 实现思路

通过对本任务的分析及相关知识学习, 制定方案如下。

- (1) 在原来 BMI 程序中显示 BMI 值后面加入多分支语句。
- (2) 多分支语句实现对 BMI 值进行区间判断。
- (3) 不同的区间显示不同的结论与建议。

### ● 实现步骤

- (1) 在 CodeBlocks 软件中打开之前实现的 BMI 程序。
- (2) 按实现思路完成本任务。

## 学习活动 4 实施实现

### 任务实现

#### ● 实现代码

- (1) 启动 CodeBlocks 软件，打开之前完成的 BMI 程序。
- (2) 打开项目中的 main.c 文件，进入编辑界面。
- (3) 在 main() 中按实现思路完成任务，代码如下。

```
int main()
{
    //定义保存数据的变量
    float height=0;
    float weight =0;
    float bmi=0;

    //输入数据
    printf("请输入你的身高（单位：米）:");
    scanf("%f",&height); //输入数据到变量中
    printf("请输入你的体重（单位：千克）:");
    scanf("%f",&weight); //输入数据到变量中
    //计算 BMI 值
    bmi=weight/(height*height);
    //显示数据
    printf("你的体质指数（BMI）为:%f\n",bmi);

    //*****以下就是本次优化的内容*****
    //判断 BMI 值的区间，给出结论和建议
    printf("-----\n");
    if(bmi<18.5)
    {
        printf("结论：你的 BMI<18.5， ");
        printf("体重过轻；属于低危险群体\n");
        printf("建议：平时多注意营养，增加体重！ ");
    }
    else if(bmi>=18.5 && bmi<24.0)
    {
        printf("结论：你的 18.5≤BMI<24， ");
        printf("体重正常；属于无风险群体\n");
    }
}
```

```
    printf("建议：别骄傲，要保持哦！");
}
else if(bmi>=24 && bmi<27)
{
    printf("结论：你的 24≤BMI<27，");
    printf("体重过重；属于低危险群体\n");
    printf("建议：该减重啦！加强运动。");
}
else if(bmi>=27 && bmi<30)
{
    printf("结论：你的 27≤BMI<30，");
    printf("轻度肥胖；属于中危险群体\n");
    printf("建议：控制饮食，加强运动，必须减重了。");
}
else if(bmi>=30 && bmi<35)
{
    printf("结论：你的 30≤BMI<35，");
    printf("中度肥胖；属于重危险群体\n");
    printf("建议：减重！减重！减重！");
}
else if(bmi>=35)
{
    printf("结论：你的 BMI≥35，");
    printf("病状肥胖；属于非常危险群体\n");
    printf("建议：在医生的指导下进行减重吧，我不能给你建议了！");
}
printf("\n-----\n");

return 0;
}
```

(4) 运行程序。

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

将所学的 if 多分支语句的知识进行综合应用，对前面任务实现的 BMI 程序进行优化，程序实现效果如图 3.15 所示。

```

请输入你的身高（单位：米）:1.7
请输入你的体重（单位：千克）:64
你的体质指数（BMI）为:22.145327
-----
结论：你的 18.5≤BMI<24，体重正常
建议：别骄傲，要保持哦！

请输入你的身高（单位：米）:1.65
请输入你的体重（单位：千克）:100
你的体质指数（BMI）为:36.730946
-----
结论：你的 BMI≥35，病状肥胖，属于非常危险群体
建议：在医生的指导下进行减重吧，我不能给你建议了！

请输入你的身高（单位：米）:1.7
请输入你的体重（单位：千克）:40
你的体质指数（BMI）为:13.840830
-----
结论：你的 BMI<18.5，体重过轻，属于低危险群体
建议：多注意营养，增加体重
    
```

图 3.15 体质指数程序实现效果

根据计算所得的 BMI 值进行区间判断。

- (1) 结论：属于哪种类型。
- (2) 建议：给出你的建议。

● 验收结果

序 号	验 收 内 容	实 现 效 果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	使用代码的规范性（变量命名、注释说明）					
3	掌握知识的情况					
4	程序性能及健壮性					
5	团队协作					

说明：在实现效果对应等级中打“√”。

● 验收评价

.....

.....

验收签字.....

## 学习活动 6 总结拓展

### 任务总结与拓展

● 实现效果

将所学的 if 多分支语句的知识进行综合应用，对前面任务实现的 BMI 程序进行优化。根据计算所得的 BMI 值进行区间判断。

- (1) 结论：属于哪种类型。
- (2) 建议：给出你的建议。

- 技术层面

if 多分支语句的应用。

BMI 值区间关系表达式的表示。

- 课程思政

通过本任务的学习，同学们完成对 BMI 程序的进一步优化。

虽然没有进行本次优化，BMI 程序也可以通过用户输入身高和体重的数据并计算出相应的 BMI 值。但这个 BMI 值对于用户来说没有什么意义，因为用户并不清楚这个值到底代表什么，需要查找相关的网页才知道其意义。

我们作为一名程序开发的初学者，除了将相关程序的知识学习好、掌握好的同时，更多地应该站在程序使用者的角度去思考，所开发的软件是不是能真正帮助用户，让用户使用起来非常方便、高效。

本次任务对 BMI 程序优化后，用户输入身高、体重后，可非常清楚地知道自己的 BMI 值处在什么区间，该区间的 BMI 属于哪个分类阶段，同时对应着自己的健康是怎样的，并且可以参考程序给出的建议。

多站在用户的角度去思考，才能编写出真正符合用户需求的软件，这是一个开发人员所必备的职业素质之一。

- 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）

---

---

---



## 分支结构学好了，BMI 程序也优化完了

### 但新问题又来了

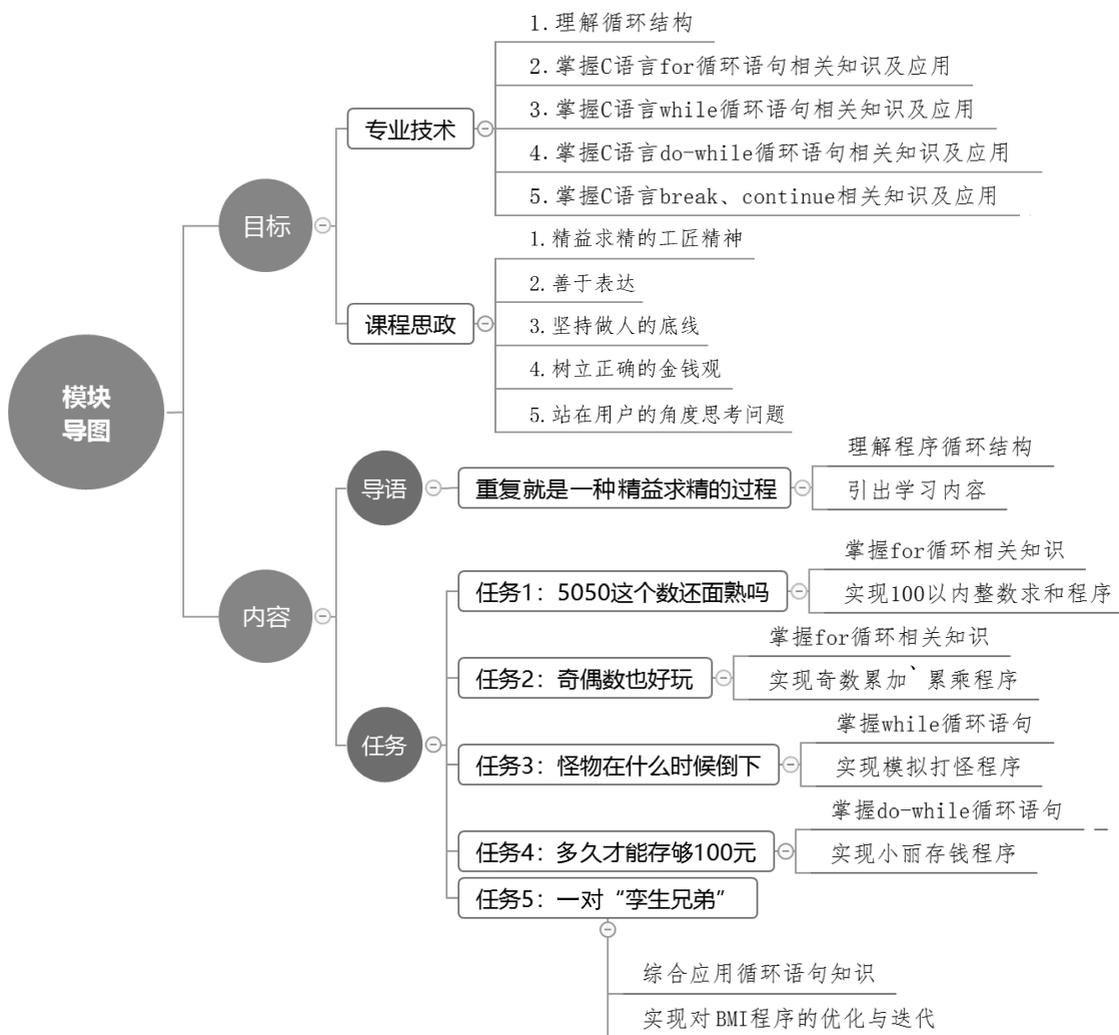
到目前为止，这个 BMI 程序运行，只能计算一个人的体质指数后就结束了

如果启动后，我们想计算多个人的体质指数该怎么办呢

（在下一个模块中就可以得到解决）

# 模块 4

## C 语言程序中的循环结构处理



## 项目导语：重复就是一种精益求精的过程



### 1. 生活中的重复

在日常生活中，我们对于重复并不陌生，其实重复就是一种精益求精的过程。

例如，同学们每天进行着的“三点一线”的行为就是一种重复。一种重复于宿舍→图书馆→食堂的生活方式，这种重复会延续 10 多年，在重复中我们收获了知识，成长了自己，这不就是对自己精益求精的表现吗？如图 4.1 所示。



#### “三点一线”的生活方式

收获知识，成长自己。

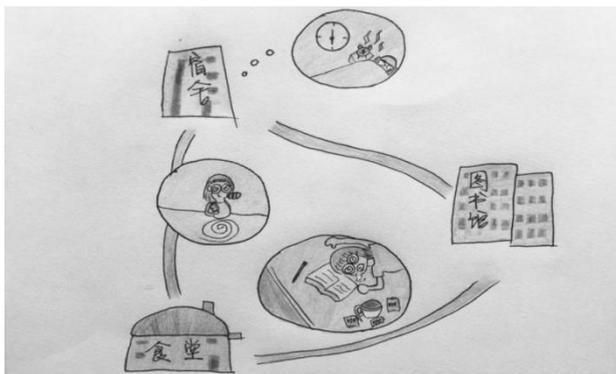


图 4.1 生活中的“三点一线”

我们很多人都经历过“刷圈”，如图 4.2 所示。通过一圈又一圈地重复，我们渐渐发现自己的体重轻了、体型好看了等，这样的重复让我们变得更加自律，不知不觉中发现这才是自己想要的自由。重复能让自己变得更好，这就是精益求精的生活方式。



#### 自律的人才会有真正的自由

可能你也经历过“刷圈”，  
一圈又一圈。



图 4.2 “刷圈”图

还有，我们很多人在自己平凡的岗位上努力着、重复着、思考着，让工艺越来越精了，让品控越来越高了，大国工匠越来越多。大国工匠就是通过一种重复，把工作做到极致的精神。

### 2. 程序中的重复

接下来，我们说说程序中的重复。

程序中的重复是指循环：

- (1) 重复执行某一段代码；
- (2) 以最终实现目标而结束；

(3) 或者永不结束。

如图 4.3 所示，当条件成立时执行循环，当条件不成立时结束。

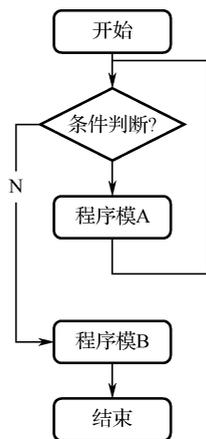


图 4.3 函数循环示意



## 来吧，正式开启程序的重复操作

为了把程序中的循环知识进行全面介绍，本模块共设计以下 5 个任务。

任务 1：5050 这个数还面熟吗。

任务 2：奇偶数也好玩。

任务 3：怪物在什么时候倒下。

任务 4：多久才能存够 100 元。

任务 5：一对“孪生兄弟”。

### 任务 1 5050 这个数还面熟吗



#### 目标描述

#### 任务描述

##### ● 编写程序实现

编写 C 语言程序，实现  $1+2+3+\dots+100$  求和。

输入：无。

处理： $1+2+3+\dots+100$ 。

输出： $1+2+3+\dots+100=5050$ 。

##### ● 技术层面

掌握 for 循环语句的相关知识。

应用 for 循环语句实现本任务。

##### ● 课程思政

精益求精的工匠精神。

## 学习活动 1 接领任务

### 领任务单

#### ● 任务确认

通过编写 C 语言程序，实现  $1+2+3+\dots+100$  求和。

输入：无。

处理： $1+2+3+\dots+100$ 。

输出： $1+2+3+\dots+100=5050$ 。

具体要求如下：

- (1) 程序最终能正确展示输出  $1+2+3+\dots+100$  之和；
- (2) 掌握 C 语言代码的使用规范（变量命名及注释说明）；
- (3) 程序能正确运行，并具有可扩展性。

#### ● 确认签字

## 学习活动 2 分析任务

编写 C 语言程序，实现  $1+2+3+\dots+100$  数相加的和，并输出结果。具体分析如下：

- (1) 1 到 100 的所有整数做相加；
- (2) 完成 100 次加法运算；
- (3) 让程序重复 100 次加法运算就可以得到结果。



如何让程序实现重复执行

## 知识学习：for 循环语句



知识学习

for 循环语句是 C 语言循环语句之一，在给定的条件成立时，会反复执行某段程序，直到条件不成立时，跳出执行循环后的代码。

### 1. 语法

```
for(表达式 1; 表达式 2; 表达式 3)
{
    语句块 1
}
...
```

说明：

- (1) for 循环语句由 3 个表达式构成，表达式之间使用分号隔开。
- (2) 表达式 1：只执行一次，用于循环变量赋初值。
- (3) 表达式 2：条件表达式，用于决定是否循环的条件，如果条件成立则循环；如果条件不成立则结束循环。
- (4) 表达式 3：记步因子，用于处理循环变量增量的表达式。

学习笔记

## 2. for 循环语句执行流程

以下程序共循环了 5 次：

```
int main()
{
    int i;//循环变量
    for(i=1; i<=10; i=i+2)
    {
        printf( "%d" ,i);
    }
}
```

程序运行后的结果为

第 1 次循环：i=1，输出：1。

第 2 次循环：i=3，输出：3。

第 3 次循环：i=5，输出：5。

第 4 次循环：i=7，输出：7。

第 5 次循环：i=9，输出：9。

执行流程如图 4.4 所示。

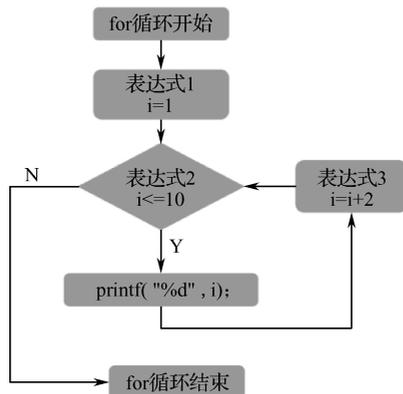


图 4.4 函数循环流程

## 学习活动 3 制定方案

### 实现本任务方案

#### ● 实现思路

通过对本任务的分析及相关知识学习，制定方案如下：

(1) 定义程序中使用到变量（2 个）。

1 个是循环变量，另 1 个是用于存放结果的变量。

(2) 利用 for 循环语句实现计算。

循环变量初始化为 1；循环条件<=100；记步因子每次+1。

在循环体内部实现 1~100 的整数相加。

(3) 输出计算结果，程序结束。

#### ● 实现步骤

(1) 在 CodeBlocks 软件中创建一个新项目，项目名称为 Sum100。

(2) 在 main() 中按实现思路编写程序代码，实现本任务。

## 学习活动 4 实施实现

### 任务实现

#### ● 实现代码

(1) 启动 CodeBlocks 软件，创建一个新的控制台项目，项目名称输入为 Sum100。

(2) 打开项目中的 main.c 文件，进入编辑界面。

(3) 在 main() 中按实现思路实现任务，其代码如下。

```
/**
 * 实现 100 以内整数的累加
 */
int main()
{
    // (1) 定义程序中使用到的 2 个变量
    int sum=0; //定义存放结果的变量，赋值为 0
    int i; //循环变量

    // (2) 利用 for 循环语句实现计算
    for(i=1; i<=100; i++)
    {
        sum=sum+i; //实现求和
    }

    // (3) 输出结果
    printf("1+2+3+...+100=%d",sum);
    return 0;
}
```

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

利用 for 循环语句实现了求 1~100 所有整数和的程序。

按制定方案进行任务实现，在正确的情况下，任务实现的效果如图 4.5 所示。

```
1+2+3+...+100=5050Press any key to continue
```

图 4.5 求 1~100 之和运行效果

● 验收结果

序 号	验收内容	实现效果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	开发环境安装与置换情况					
3	掌握知识的情况					
4	程序运行情况					
5	团队协作					

说明：在实现效果对应等级中打“√”。

● 验收评价

.....

.....

验收签字.....

## 学习活动 6 总结拓展

### 任务总结与拓展

● 实现效果

利用 for 循环语句实现了求 1~100 所有整数和的程序。

● 技术层面

for 循环语句是实现程序重复执行某一段程序块的方式之一。

1. 语法

```
for(表达式 1; 表达式 2; 表达式 3)
{
    程序块
}
```

说明：

for 是命令动词，后面紧跟一对小括号，小括号中写 3 个表达式，表达式之间使用“;”间隔；

{ } 内是循环执行的程序代码。

执行流程如下：

(1) 执行表达式 1：给循环变量赋初值；

(2) 执行表达式 2: 判断条件是否成立。如果成立则执行循环; 如果不成立则结束循环;  
(3) 当程序执行到 for 循环的 “}” 时, 先返回执行表达式 3, 再执行表达式 2。  
以此类推, 从而实现循环执行程序的目的。

## 2. 本任务的代码

```
int main()
{
    //定义存放结果的变量, 赋值为 0
    int sum=0;
    int i; //循环变量

    //循环实现求和
    for(i=1;i<=100;i++)
    {
        sum=sum+i;
    }
    //输出结果
    printf("1+2+3+...+100=%d",sum);

    return 0;
}
```

### 说明:

循环变量 i: 初值为 1, 只要小于或等于 100, 则条件成立, 进入循环体执行求和。

求和变量 sum: 第一次循环时, sum=0, i=1。先计算 sum+i, 则结果为 1, 然后再赋值给 sum, 则第一次循环后, sum=1;

循环返回执行 i++, 则 i=2; 满足条件, 继续执行循环, 先计算 sum+i, 则结果为 1+2=3, 然后再赋值给 sum, 则第二次循环后, sum=3;

循环返回执行 i++, 则 i=3; 满足条件, 继续执行循环, 先计算 sum+i, 则结果为 1+2+3=6, 然后再赋值给 sum, 则第三次循环后, sum=6; 依此一直执行 100 次循环, 实现了 1+2+3+...+100 的求和。

当 100 次循环后, i+=101 时, 条件不成立了, 结束循环。执行循环外的代码, 则执行输出结果的代码, 程序最终结束。

### ● 课程思政

根据实现的思路, 设计好程序代码, 通过 100 次重复执行, 最终达成本任务的目标。在日常生活中, 重复就是一种精益求精的表现, 我国有那么多的大国工匠, 他们对自己的工作, 就是日复一日, 年复一年地重复着, 实现从理解到熟悉再到升华的过程。

对我们每一个人来说, 如果:

坚持每天做 1 分钟的俯卧撑;

坚持每天读 10 分钟的英文;

坚持每天写 30 分钟的程序;

.....

这就是对自己精益求精的一种表现, 只要一直坚持下去, 一定会有收获。

● 教学拓展

通过本任务的学习与实现，同学们初步掌握了 for 循环的应用，试着编写 1~10 以内的整数相乘。

● 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）

.....

.....

.....



**还想玩玩**

**下一回：奇偶数也好玩**

（巧妙对 for 进行设计，会有意想不到的效果）

**任务 2 奇偶数也好玩**



**目标描述**

**任务描述**

● 编写程序实现

编写 C 语言程序实现 2~100 的所有偶数和。

输入：无

处理：2+4+6+...+100

输出：2+4+6+...+100=?

● 技术层面

掌握 for 循环语句的相关知识。

掌握程序判断奇偶数的方法。

应用 for 循环语句实现本任务。

● 课程思政

善于表达。

**学习活动 1 接领任务**

**领任务单**

● 任务确认

编写 C 语言程序实现 2~100 的所有偶数和。

具体要求如下:

- (1) 程序最终能正确展示输出结果;
- (2) 掌握 C 语言代码的使用规范 (变量命名及注释说明);
- (3) 程序能正确运行, 并具有可扩展性。

- 确认签字

## 学习活动 2 分析任务

编写 C 语言程序, 实现对奇偶数的判断, 并结合 for 循环完成任务, 具体分析如下:

- (1) 偶数是指能被 2 整除的数, 如 0, 2, 4, 6, 8, 10 等。
- (2) 奇数是指不能被 2 整除的数, 如 1, 3, 5, 7, 9, 11 等。



### 如何判断奇偶数

#### 知识学习: 程序中奇偶数的处理



#### 知识学习

在程序中, 我们该如何判断奇偶数呢?

使用取余 (%) 运算就可以确定某一个数是不是偶数。

假如要判断的这个数是  $x$ , 那么

$x \% 2$ , 如果结果为 0, 则说明该数是偶数, 否则为奇数。

如:

$$4 \% 2 = 0$$

$$10 \% 2 = 0$$

$$6 \% 2 = 0$$

说明: 4, 10, 6 为偶数。

$$3 \% 2 = 1$$

$$7 \% 2 = 1$$

$$11 \% 2 = 1$$

说明: 3, 7, 11 为奇数。

学习笔记

## 学习活动 3 制定方案

### 实现本任务方案

- 实现思路

通过对本任务的分析及相关知识学习, 实现方案如下。

方法一: 采用取余 (%) 实现

- ① 定义程序中使用的变量 (2 个)。

它们是循环变量和存放结果的变量。

- ② 编写 for 循环，初始化循环变量，设置好条件和记步因子。
- ③ 在循环体内判断使用取余 (%) 运算是不是偶数，如果是偶数则相加。
- ④ 输出结果，程序结束。

**方法二：采用对 for 循环记步因子实现**

- ① 定义程序中使用的变量（2个）。
- 它们是循环变量和存放结果的变量。
- ② 编写 for 循环，初始化循环变量=0，设置好条件和记步因子为每次+2。
  - ③ 在循环体内直接求和。
  - ④ 输出结果，程序结束。

#### ● 实现步骤

- (1) 在 CodeBlocks 软件中创建一个新项目，项目名称为 oushu100。
- (2) 在 main() 中按实现思路编写程序代码，以实现本任务。

## 学习活动 4 实施实现

### 任务实现

#### ● 实现代码

- (1) 打开 CodeBlocks 软件，创建一个新的控制台项目，项目名称输入为 oushu100。
- (2) 打开项目中的 main.c 文件，进入编辑界面。
- (3) 在 main() 中按实现思路完成任务，其代码如下。

**方法一：采用取余 (%) 实现**

```
//方法一参考代码
int main()
{
    //定义变量
    int i; //循环变量
    int sum=0; //保存结果的变量

    //实现偶数相加的方法一
    for(i=1 ; i<=100 ; i++)
    {
        if( i%2==0) //判断是不是偶数
        {
            sum=sum+i;
        }
    }
    printf("方法一： 2+4+6+...+100=%d",sum);
}
```

### 方法二：采用对 for 循环记步因子实现

```
//方法二参考代码
int main()
{
    //定义变量
    int i; //循环变量
    int sum=0; //保存结果的变量

    //实现偶数相加的方法二

    for(i=0 ; i<=100 ; i=i+2) //循环变量初始=0，记步因子：i=i+2
    {
        sum=sum+i; //没有判断，直接求和
    }
    printf("方法二：2+4+6+...+100=%d",sum);
}
```

#### 思考：

本任务通过上面两种方法都可以实现，请问哪种方法更优呢？并说明理由。

## 学习活动 5 测试验收

### 任务测试验收单

#### ● 实现效果

利用 for 循环语句实现了求 100 以内（含 100）所有偶数和的程序。

按制定的方案进行任务实现，在正确的情况下，任务实现的效果如图 4.6 所示（以方法一为例）。

方法一：2+4+6+...+100=2550 Press any key to continue

图 4.6 求 100 以内（含 100）的偶数及任务运行效果

#### ● 验收结果

序号	验收内容	实现效果				
		A	B	C	D	E
1	任务要求的功能实现情况					
2	使用代码的规范性（变量命名、注释说明）					
3	掌握知识的情况					
4	程序性能及健壮性					
5	团队协作					

说明：在实现效果对应等级中打“√”。

● 验收评价

-----  
-----

验收签字\_\_\_\_\_

## 学习活动 6 总结拓展

### 任务总结与拓展

● 实现效果

利用 for 循环语句实现了求 100 以内（含 100）所有偶数和的程序。

● 技术层面

如果某一个数%2（取余）运算如果为 0，则说明这个数是一个偶数。巧设 for 循环语句的各表达式，以实现不同的目的，如使用两种方法实现本任务的目标。

方法一：循环变量 i 初值为 1，每循环一次+1，一直循环到 100。

每次循环都要判断 i%2 是否为 0。

如果为 0 则求偶数和，否则（奇数）不求和。

方法二：循环变量 i 初值为 0，每循环一次+2，一直循环到 100。

这样设计，则 i 的值就是 0, 2, 4, 6, ..., 100，所以不用判断，直接求和。

● 课程思政

方法一和方法二都能达成本任务的目标，哪种方法更优呢？

如果让你来回答，应该如何表达呢？

回答示例参考。

各位同学好！老师好！

我叫×××，通过小组交流后，我们认为方法二更优，原因如下。

(1) 从循环的次数来说：方法一共循环 100 次，方法二共循环 51 次，方法二胜出；

(2) 从消耗时间来说：方法一每次都要除以 2 求余，要消耗一定时间，而判断也要消耗一定的时间。而方法二，不做求余，也不用判断，直接做加法运算，方法二胜出。

综上所述，我们小组认为方法二更优！

谢谢大家！

大家觉得按以上方式进行表达，效果是不是很好。

在社会生活中，沟通表达能力是非常重要的。我们每个人都应该具备思考与表达自己想法的能力，通过不断地进行重复演练，逐步培养自己的表达与沟通的能力。

● 教学拓展

通过本任务的学习，尝试利用 for 循环语句，编写实现  $1!+2!+3!+4!+5!+6!+7!+8!+9!+10!$ 。

说明：n!即求 n 的阶乘，如  $2!=1*2$ ； $5!=1*2*3*4*5$ 。

- 任务小结（请在此记录你在本任务中对所学知识的理解与实现本任务的感悟等）

---

---

---

---

---



## for 循环常用于明确循环次数的情况

在现实中，遇到不确定的情况该怎么办呢

**下一回：怪物在什么时候倒下**

（解决不确定性）

### 任务 3 怪物在什么时候倒下



#### 目标描述

##### 任务描述

- 编写程序实现

编写 C 语言程序，实现模拟游戏中“打怪”的过程。

设定怪物生命值（假定），从键盘输入一个整数作为攻击力，用怪物生命值减去攻击力，当怪物生命值 $\leq 0$ 时，程序结束。

- 技术层面

掌握 while 循环语句的相关知识。

理解死循环的概念。

应用 while 循环语句实现本任务。

- 课程思政

坚守做人的底线。

## 学习活动 1 接领任务

### 领任务单

#### ● 任务确认

编写 C 语言程序，实现模拟游戏中“打怪”的过程。

具体要求如下：

- (1) 程序最终能正确展示“打怪”结果；
- (2) 掌握 C 语言代码的使用规范（变量命名及注释说明）；
- (3) 程序能正确运行，并应具有可扩展性。

#### ● 确认签字

## 学习活动 2 分析任务

编写 C 语言程序，实现模拟游戏中“打怪”的过程。

- (1) 设定怪物的生命值，并且该值是固定的。
- (2) 使用循环来实现攻击的过程。
- (3) 输入攻击值，并用怪物生命值减去攻击值。
- (4) 条件判断怪物生命值是否 $\leq 0$ 。
- (5) 如果不成立，则继续循环输入攻击力“打怪”。
- (6) 如果成立，则结束循环，显示“Game Over!”和总攻击次数。



**不同的攻击力，循环次数不一样**

## 知识学习：while 循环语句



## 知识学习

while 循环语句是 C 语言循环语句之一，在给定的条件成立时，反复执行某段程序，直到条件不成立时，跳出循环执行循环后的代码。

### 1. 语法

```
while(循环条件)
{
    程序语句
}
...
```

说明：

- (1) while 语句可以实现循环结构程序，{}内就是循环执行的程序代码；
- (2) 先判断，再执行；
- (3) 如果循环条件成立，则执行循环，否则结束循环，执行循环后的语句。

### 学习笔记

-----

-----

-----

-----

-----

-----

-----

-----

## 2. while 循环语句执行流程 (见图 4.17)

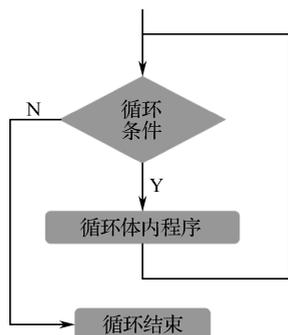


图 4.7 while 循环语句执行

## 3. while 循环语句举例

实现 100 以内的偶数之和。

```

int main()
{
    //定义变量
    int i; //循环变量
    int sum=0; //保存结果的变量

    i=0;
    while(i<=100)
    {
        sum=sum+i;
        i=i+2;
    }
    printf("2+4+6+...+100=%d",sum);
}
    
```

## 4. 死循环

死循环是指永远不停止的循环，也就是循环条件永远为真。

真正的死循环是没有什么意义的，但我们可以利用这个特性来设计程序，如不明确真实的循环次数时，就可以在循环中设定循环结束的条件，即只要条件成立时就可结束死循环了。

在 C 语言中，循环条件成立，即是真，使用 1 来表示。所以将 while 的循环条件设置为 1 时，就是一个死循环的设计。

```

while(1)
{
    程序语句
}
    
```