高等学校计算机专业教材精选·算法与程序设计

C语言程序设计
——案例驱动教程

刘玉英 主编
刘玉英 刘臻 肖启莉 编著

清华大学出版社

高等学校计算机专业教材精选·算法与程序设计

C语言程序设计
——案例驱动教程

刘玉英 主编
刘玉英 刘臻 肖启莉 编著

清华大学出版社
北京

内 容 简 介

本书通过具有实用性和趣味性的案例引出相关知识点，介绍知识点，强化学习知识点，总结应用知识点。通过案例学习理论知识，模仿改写程序，启发引导读者把数学思想转换成用C程序代码来表现，即编写程序，提高知识的掌握水平以及应用能力。

本书具有覆盖面广、案例丰富、突出案例驱动的特色；详略得当、主次分明，在主要知识点上下工夫，不面面俱到；设计了“请思考”，启发引导读者进行更深入的探讨，举一反三。对于容易出现的错误以及需要注意的事项，设计了温馨提示以提醒读者，避免学习中走弯路。为了配合本书的学习，在附录中还提供了两套自测练习题及其参考答案。

本书适用于C语言程序设计的初学者，可以作为普通高等院校电子信息类专业程序设计基础的教材，也可作为有兴趣学习C语言的其他专业学生的教材，同时也适合自学。

图书在版编目（CIP）数据

C语言程序设计——案例驱动教程 / 刘玉英主编. —北京：清华大学出版社，2011.9
（高等学校计算机专业教材精选·算法与程序设计）
ISBN 978-7-302-26025-7

Ⅰ. ①C… Ⅱ. ①刘… ②刘… ③肖… 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(2011)第176652号

责任编辑：张　民　王冰飞
责任校对：梁　毅
责任印制：何　芊

出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**邮　　编**：100084
社　总　机：010-62770175　　**邮　购**：010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：清华大学印刷厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185×260　　**印　张**：22.75　　**字　数**：567千字
版　　次：2011年9月第1版　　**印　次**：2011年9月第1次印刷
印　　数：1～4000
定　　价：33.00元

产品编号：036226-01

出版说明

我国高等学校计算机教育近年来迅猛发展，应用所学计算机知识解决实际问题，已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在，很多高等学校都在积极探索符合自身特点的教学模式，涌现出一大批非常优秀的精品课程。

为了适应社会的需求，满足计算机教育的发展需要，清华大学出版社在进行大量调查研究的基础上，组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材，把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生，使学生通过本套教材，学习计算机科学与技术方面的基本理论和基本知识，接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各高校相应课程的主讲，有一定经验积累，且编写思路清晰，有独特的教学思路和指导思想，其教学经验具有推广价值。本套教材中不乏各类精品课配套教材，并努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则，书中的许多必须掌握的理论都将结合实例来讲，同时注重培养学生分析问题、解决问题的能力，满足社会用人要求。

(4) 易教易用，合理适当。本套教材编写时注意结合教学实际的课时数，把握教材的篇幅。同时，对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答，学生上机实验指导、教学网站等辅助教学资源，方便教学。

随着本套教材陆续出版，我们相信它能够得到广大读者的认可和支持，为我国计算机教材建设及计算机教学水平的提高，为计算机教育事业的发展做出应有的贡献。

清华大学出版社

前　言

C语言是国内外广泛使用的一种计算机语言，在计算机编程语言的发展史上，占据着极其重要的地位，无论是计算机程序开发人员，还是非计算机专业人员，掌握面向过程程序设计仍然是计算机工作者的基本功，并且几乎所有的计算机学科都把C/C++语言当做最基础的科目之一。C语言是一门极为重要的专业基础课程，今天我们学习C语言正是为今后的学习、工作打下专业基础。

教育部高等学校计算机科学与技术教学指导委员会专家曾经指出，计算机教育的四个方向（计算机科学方向、计算机工程方向、软件工程方向、信息技术方向）对于程序设计基础都有较高的要求，因为它是所有后续课程的专业基础。用C语言作为计算机程序设计的入门语言，要正确处理算法与语法的关系，学习中不应该把重点放在语法规则上，而是要放在解题的思路上，通过大量的例题学习怎样设计一个算法，构造一个程序；语法虽然重要，但不能在语法细节中死抠。学习的重点是从程序入手，模仿编程，进而逐步深入，自己推敲好的算法，自行设计调试程序，通过程序的学习掌握C语言的主要知识点。程序设计是一门实践性很强的课程，既要掌握概念，又要动手编程，还要上机调试运行，这三方面配合得当才能收到好的学习效果。

本书希望通过具体案例引出相关知识点，介绍知识点，强化学习知识点，总结应用知识点。通过案例学习理论知识，模仿改写程序，启发引导读者自主编写程序，提高知识的掌握水平以及知识应用能力。本书具有如下特色：

（1）覆盖面广，突出案例驱动特色。

在对C语言的主要知识点分析归纳的基础下，精选每个部分的案例。对于每个具体案例，都从知识点出发，分析问题的解决方法，然后编写出程序代码。

（2）详略得当，主次分明。

在主要知识点上下工夫，不可能面面俱到，必须有所取舍。对于非重点或较复杂的内容略讲，如数组中重点是一维数组、字符数组；在结构与联合中，重点讲解结构，而联合的内容重点在于与结构的区别。

（3）案例生动，实用性强。

本书针对C语言特点，精选重点，强化主要概念，图文并茂地讲解每个重要知识点，并配以较多容易理解的程序实例，以例题释含义、总结出规律，便于理解和应用。同时在每一章的主要内容讲解之后，充分利用前面的知识，将多个知识点有机地结合起来，设计了有一定难度并且趣味性强的综合应用实例，以加强对所学知识的理解和运用，如置换问题、鸡兔同笼问题、发纸牌游戏、随机给儿童出加法测验题、竞赛评分、小孩分糖果、约瑟夫问题、利用随机数生成函数计算圆周率、求若干个正整数的最小公倍数、用古典筛法求素数、古代处决犯人问题、验证卡布列克常数、用位运算的方式交换两个变量的值等。由浅入深地讲述，生动形象的程序实例，使读者学起C语言来有兴趣，不再感觉学习是很难、很枯燥的事了。

(4) 设计“思考”,启发动脑。

在典型例题之后,设计了思考题,启发引导读者进行更深入的思考,举一反三。不少读者反映,自己的编程能力差,案例程序可以读懂,但是却不会自己编写程序。作者期望通过设计思考题的方式引导读者增强编程能力。对于容易犯的错误以及需要注意的事项,设计了温馨提示,以润物细无声的方式提醒读者,避免学习中走弯路。

本书适用于C语言程序设计的初学者,可以作为普通高等院校电子信息类专业程序设计基础的教材,也可作为有兴趣学习C语言的非计算机专业学生的教材,同时本书也适合自学。

全书共11章,由刘玉英给出写作提纲和基本要求。第1、7、9、10章由刘玉英编写,第2、3、4章由肖启莉编写,第5、6、8章由刘臻编写,第11章由三人共同完成,附录部分由刘玉英编辑整理。最后全书由刘玉英统编定稿。

尽管本书作者都是多年讲授C语言程序设计课程的教师,有着比较丰富的教学经验,但是由于受到水平和写作时间的限制,仍然可能存在这样或那样的不足之处,恳请使用本书的教师、学生和其他读者批评指正,以便修改。

我们的联系方式是E-mail: liuyy@zwu.edu.cn。

作　者

2011年5月

目　录

第1章　C语言知识初步

本章主要内容：

- 简述C语言的发展；
- 通过C语言的实例介绍C语言的特点；
- 算法与流程图；
- C语言程序的开发步骤。

1.1　概　　述

语言是人与人之间交流的工具，如汉语、英语等。C语言是人与计算机交流的工具。计算机最基本的功能是计算，如何实现计算，需要人向计算机发送指令。用C语言可以向计算机发送进行某种运算（或操作）的指令。指令（在C语言中称为语句）的有序集合即为程序，所以C语言又被称为程序设计语言。

C，是一种通用的程序设计语言，主要用来进行系统程序设计。它具有高效、灵活、功能丰富、表达力强和移植性好等特点，在程序员中备受青睐。C语言至今仍是目前世界上流行、使用最广泛的高级程序设计语言。

对操作系统、系统使用程序以及需要对硬件进行操作的场合，用C语言编写的程序明显优于其他高级语言编写的程序，许多大型应用软件都是用C语言编写的。

在C语言诞生以前，操作系统及其他系统软件主要是用汇编语言实现的。由于汇编语言程序设计依赖于计算机硬件，其可读性和可移植性都很差，而一般的高级语言又难以实现对计算机硬件的直接操作，因此人们需要一种兼有汇编语言和高级语言特性的语言。C语言就是在这种环境下产生的。

C语言最早是由Dennis Richie于1973年设计并实现。它的产生同UNIX系统之间具有非常密切的联系——C语言是在UNIX系统上开发的。而无论UNIX系统本身还是其上运行的大部分程序，都是用C语言编写实现的。同时，C语言同样适合编写不同领域中的大多数程序。C语言已经成为全球程序员的公共语言，并且产生了当前两个主流的语言C++和Java，它们都建立在C语言的语法和基本结构的基础上，而且现在世界上的许多软件都是在C语言及其衍生的各种语言的基础上开发而成的。

目前，在微机上广泛使用的C语言编译系统有Turbo C、Win-TC、C-Free、Visual C++ 6.0、C++ Builder 6.0等。虽然它们的基本部分都相同，但使用时有一些差异，本书采用Microsoft Visual C++ 6.0作为上机编程实验环境。

1.2 认识C语言程序

首先来看几个C语言的简单程序实例，通过程序来初步了解C语言。

【案例1.1】 输出简单的字符串“Hello, World!”。

```
#include<stdio.h>                          /*编译预处理命令*/
void main()                                /*main是C语言程序的主函数名*/
{
    printf("Hello, world!");               /*printf()是输出函数*/
}
```

案例1.1是一个完整的C语言程序。第一行是编译预处理命令，用“#”开头，“include”表示文件包含，尖括号中就是被包含的文件，“stdio.h”表示输出输入的头文件，该文件是由编译系统提供的，对输入输出函数提供支持。该行完整的含义是将输入输出的头文件包含到本程序中来。

第二行中，main是C语言程序的主函数名，在一个完整的程序中有且只有一个main函数。程序执行时从main开始，到右花括号结束。void表示函数的返回类型为无返回值。第二行也被称为函数首部。

第三行中，printf()是输出函数，该函数是系统提供的标准库函数，用户直接调用就可以了，使用中有一些具体要求，如输出格式符、控制字符等，详见第2章的叙述。在这里是用来输出双引号中的多个字符。双引号中的若干个字符被称为字符串。本行被称为函数调用语句。

程序中“{”和“}”之间的部分称为函数体，也就是一个程序所需要完成的任务，用若干条语句实现，写在函数体内。

程序中“/*”和“*/”中间的部分是注释，用于解释程序或语句的作用，不参与程序的编译与执行。本案例程序的运行结果为：

```
Hello, World!
```

【案例1.2】 求两个整数的和，并显示输出。

分析：求两个数之和，在程序中要先定义两个整型变量，设它们为x、y，然后把数据存放在变量中，求它们的和，并把它存放在另一个变量z中，跟数学算式$z=x+y$的含义是一样的。

```
#include<stdio.h>
void main()
{   int x, y, z;                           /*定义整型变量x、y、z*/
    x=3;  y=5;                             /*为变量x赋值为3，为变量y赋值为5*/
    z=x+y;                                 /*求x与y的和，并将和赋予变量z*/
    printf("z=%d\n",z);                    /*输出结果*/
}
```

案例1.2在函数体内首先定义了3个整型变量，int是定义整型变量的关键词，变量名

分别为 x，y，z，为变量 x、y 分别赋值为 3 和 5，再计算它们的和将其赋给变量 z，最后按照函数 printf()中指定的格式输出结果。双引号“z＝%d/n”中，z＝是普通字符，输出时原样输出，%d 是格式控制符，表示在该位置输出一个十进制整型数，其数值由逗号后面的变量 z 确定。程序的运行结果为：

```
z=8
```

程序中第三行定义变量也称为声明语句，第四至六行为执行语句。声明语句不产生机器操作，只是通知编译系统定义变量的类型以及变量名称，便于系统在内存中分配相应的内存单元，并在以后的执行中按照指定名称和类型进行相关处理；执行语句产生机器操作，要完成某种指定任务，例如赋值、算术运算、输入、输出等。

相关知识点 1：标识符的命名规则

每个变量都要有一个合法的名称，以便在编程时使用。名称也被统称为标识符，它可以是变量名、数组名、函数名、指针变量名等。标识符可以由单个英文字母构成，如 x、y 等，也可以是英文字母与数字的组合，如 x2，还可以是下划线带字母或数字，如_123 等。但是，标识符的第一个字母只能是字母或下划线。C 语言中的关键字（如 int、char、for 等）不能作为标识符来使用，因为它们有特定的含义。

【案例 1.3】 顺序输出 26 个英文小写字母。

分析：为了输出 26 个英文字母，先输出 a，再输出 b，接着输出 c，按照字母的顺序输出方便处理。26 个英文字母在 ASCII 码表（在附录 C 中）上就是按顺序连续排列的，它们的 ASCII 码值后面的一个比前面一个大 1。那么，先给出第一个字符，加 1 就是下一个字符的值。找到了这个规律，用循环语句输出 26 个英文字母非常方便。

```
#include<stdio.h>
void main()
{   char c;                          /*定义字符型变量 c*/
    int i;                           /*定义整型变量 i*/
    c='a';                           /*为字符变量 c 赋值为'a', 即输出的第一个字母*/
    for(i=0;i<26;i++)                /*循环语句 for, 循环 26 次*/
    {   printf("%c ",c);             /*输出一个英文字母*/
        c=c+1;                       /*为输出下一个英文字母作准备*/
    }
}
```

程序运行结果为：

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

案例 1.3 采用的方法是每次输出一个字母，输出 26 次完成任务。首先将第一个要输出的字母‘a’赋予字符变量 c，输出后改变 c 的值，使 $c=c+1$ 后再输出。‘a’的 ASCII 值为 97，加 1 后为 98，98 也是字符‘b’的 ASCII 值，依此类推，c 变量进行 26 次加 1 后输出，即把从 a 到 z 的 26 个字母输出完毕。

在程序中，‘a’是字符常量，而 c 是字符变量。

for()语句是循环语句，循环次数用变量 i 来控制，变量 i 的值从 0 开始，到 25 结束，每次加 1，共进行 26 次循环。

为了在输出时使字符与字符间隔开，在输出一个字符后加上一个空格，在程序中由"%c "控制。

请思考：若输出 26 个大写英文字母，程序应如何改？

温馨提示：

① 字符常量带单引号，字符变量没有单引号。

② 所有变量必须先定义，后使用。定义即是声明变量的类型、变量名等，使用可以是为变量赋值(如 c='a')，或进行运算(如 $c=c+1$)等。

相关知识点 2：C 语言中的关键字

关键字是由 C 语言预定义的具有特定含义的单词，通常也称为保留字，它们在程序中有特定的使用目的，在定义标识符时不要使用这些关键字。C 语言共有 32 个关键字，主要分为以下几类。

数据类型(12 个)：char、double、enum、float、int、long、short、signed、struct、unsigned、union、void

存储类型(4 个)：auto、static、extern、register

控制语句(12 个)：break、case、continue、default、do、else、for、goto、if、return、switch、while

其他关键字(4 个)：const、sizeof、typedef、volatile

在编写程序时，关键字只能是小写字母，若改为大写字母就不再是关键字，而是普通标识符。

相关知识点 3：C 语言中的语句

C 语言的语句用来向计算机系统发送操作指令。C 语言程序由函数构成，而函数由若干条语句按照一定的次序构成。按照语句在程序中所起的作用可分为 5 类：

(1) 控制语句。控制语句可完成一定的控制功能，例如分支、循环、流程转向控制等。C 语言有 9 种控制语句，具体如下。

if-else：条件语句，用来实现选择结构。

switch：多分支选择语句。

for：循环语句，用来实现循环结构。

do-while：循环语句，用来实现循环结构。

while：循环语句，用来实现循环结构。

continue：流程控制语句，结束本次循环。

break：流程控制语句，终止执行 switch 或循环语句。

return：流程控制语句，从函数返回。

goto：流程控制语句，转向语句，现在已基本不用。

(2) 表达式语句。它由一个表达式加分号构成。例如：

```
z=x+y
```

是一个表达式，而

```
z=x+y;
```

是表达式语句，它们只差一个分号，最简单的表达式语句是赋值表达式语句。再次强调：语句以分号结尾。

(3) 函数调用语句。它由函数调用加分号构成，例如：

```
printf("Hello, world!");
```

printf()是一个库函数，上面语句是调用该函数。函数调用也可以出现在表达式语句或其他场合中。例如：

```
c=max(a,b);                          /* 表达式语句 */
```

或

```
printf("%d",max(a,b));               /* 函数调用语句 */
```

都正确。

(4) 空语句。只由一个分号构成，它什么操作都不执行。空语句有时用做流程的转向点，有时也可作为循环体语句，只循环规定次数，但是任何实际操作都不做。

(5) 复合语句。用花括号括起来的若干语句构成复合语句，经常在 if-else 或循环中使用复合语句。

相关知识点 4：C 语言程序结构特点

(1) C 语言程序由函数构成，而函数由若干条语句构成，每个 C 语句都由分号结尾。

(2) C 语言程序可以由若干个函数构成。但是，main 函数必须有，而且只能有一个。除 main 函数外，还可以包含标准库函数和自定义的函数。

(3) C 语言的特点：

① 语言简洁，结构紧凑，使用方便、灵活。

C 语言一共有 32 个关键字和 9 条控制语句，且源程序书写格式自由，在一行中可以写一条语句，也可以写多条语句。一条语句也可以写在多行中。

② 运算符极其丰富，数据处理能力强。

C 语言一共有 45 种运算符，它把括号、赋值号、强制类型转换符号等都作为运算符处理，使得 C 语言的运算符类型极为丰富，表达式类型多样化。灵活使用可以实现其他高级语言难以实现的运算和操作。

③ 数据结构丰富。

C 语言的数据类型有整型、实型、字符型、数组、指针、结构、联合等，用它们可以实现各种复杂的数据结构(如链表、树等)。特别是指针类型，使用起来灵活多变。

④ 具有结构化的控制语句，是一种模块化的程序设计语言。

如 if-else 语句、while 语句、for 语句、break 语句等，可以在程序中使用所有的控制语句构成分支结构、循环结构。另外，函数是 C 语言的基本单位，用函数作为程序模块，以实现程序的模块化。

⑤ 可移植性好。

C 程序本身基本上可以不做任何修改，就能运行在各种不同型号的计算机和各种操作系统环境上。例如，现在常用的 TC、Visual C++ 6.0 等编译系统。

⑥ C 语言允许直接访问物理地址，可以直接对硬件进行操作。

C 语言提供了某些接近汇编语言的功能，如位运算等，能直接访问物理地址，直接对硬件进行操作，从而有利于编写系统软件。

⑦ C 语言程序生成目标代码质量高，程序执行效率高。

⑧ C 语法限制不太严格，程序设计自由度大。

一般的高级语言语法检查比较严，能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。

1.3 算法与流程图

程序是为了实现某种运算或操作，因此程序一般包括对数据的描述和对操作的描述。

数据一般分为整型、实型、字符型等基本数据类型，数据的存储方式又分为静态存储和动态存储等。对数据的描述就是要在程序中指出数据的类型和存储方式，以便对数据进行相应的操作。对操作的描述，就是对数据的处理步骤，即每一步要做的事情，也称为算法。

算法是指为解决一个问题而采取的方法和步骤，编程序就是把方法或步骤用一种编程语言来描述。算法实际上是解决“做什么”和“怎么做”的问题。算法具有以下特点：

(1) 有穷性。一个算法必须保证执行有限步之后结束。

(2) 确定性。算法的每一步骤必须有确切的定义。

(3) 输入。一个算法有 0 个或多个输入，以刻画运算对象的初始情况，所谓 0 个输入是指算法本身决定了初始条件，不需要从键盘输入数据。

(4) 输出。一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

(5) 有效性。算法原则上能够有效地运行，并能得到确切的结果。

解决一个问题可以有多种算法，不同的算法可能所需要的时间(即运行速度)和空间(即占用内存单元多少)都不同，效率也不同，算法的优劣程度也不同。算法可以用自然语言描述，可以用伪代码描述，也可以用流程图描述。

流程图是指用一些图框和流程线来表示操作步骤。美国国家标准化协会 ANSI 规定了一些常用的流程图符号，如图 1.1 所示。

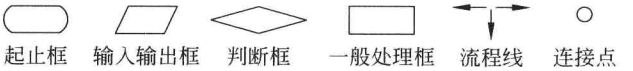

图 1.1 常用流程图符号

【案例 1.4】 求 1～100 之间的偶数之和，并显示输出结果。

分析：求 1～100 之间的偶数之和，即 2＋4＋6＋…＋98＋100，有 50 个数相加，因为要多次求和，运算步数较多，所以用循环处理比较简单。先定义两个变量，一个变量 i 用来做循环控制变量，初始值为 0，下次使之加 2，变成 2，再加 2 变成 4，依次变成 6、8、10、…、100 为止，另一个变量 sum 用于累加，初始值为 0，在 i 变化的同时累加 i 的值，每次使 $sum+i$ 赋予 sum，首次累加 $sum=0$，第 2 次累加 $sum=2$，第 3 次累加 $sum=6$，直至累加到 $i=100$ 结

束，最后输出累加和。

从自然语言描述的算法中可以看出，变量 i 需要变化多次，从 0 到 2，再到 4，…，直至 100，变量 sum 同样变化多次，随 i 的变化累加，这个过程由循环完成。

求 1～100 之间的偶数之和的算法描述用流程图来表示更加简单明了，如图 1.2 所示。

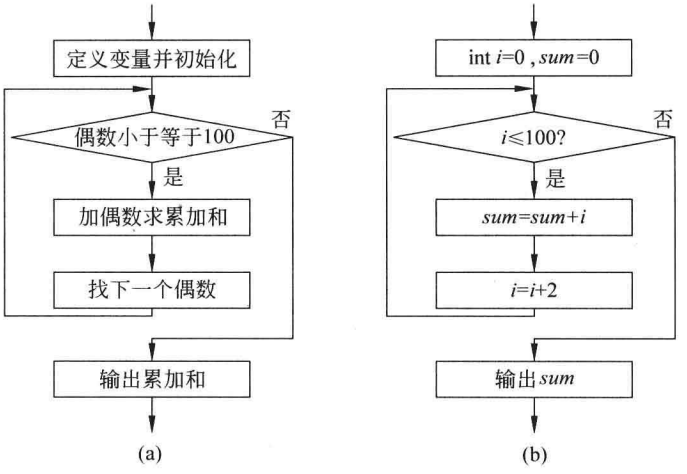

图 1.2　求 1～100 之间偶数和流程图

图 1.2(a)是用自然语言描述的框图表示，图 1.2(b)是同一问题的符号描述，(a)与(b)相同位置的图框含义相同。按照(b)图的流程顺序可以方便地编写出一个解决该问题的程序。

1.4　C 语言程序的开发

C 语言程序的开发分以下几步来完成：

(1) 首先要明确任务，也就是了解要解决的问题是什么。

(2) 设计问题的解决方案，也就是确定算法。

(3) 编写程序代码，用 C 语言的语句按照确定的算法写出程序。

(4) 在编译环境中将 C 程序源代码录入到计算机中，并存盘，该过程即为编辑。

(5) 进行编译、连接，排除程序中存在的语法和词法错误。

(6) 运行程序并用数据进行测试，检查是否能够完成预定任务。若程序的执行结果满足题意要求，那么，这个程序就开发成功了。

现在以求 1～100 之间的偶数之和为例简单描述一下开发过程。

1. 编写源程序

任务：求 1～100 之间的偶数之和。

算法：用循环控制求累加和，反复执行 $i=i+2$，$sum=sum+i$ 操作。

按照图 1.2(b)所示的流程图写出源程序：

```
#include<stdio.h>
void main()
```