



“十三五”职业教育国家规划教材

C 语言程序设计

主 编 张成叔 万 芳

智慧职教

MOOC+SPOC课程



扫描二维码
了解本书的
配套资源

- ▲ 微课视频
- ▲ 教学课件PPT
- ▲ 课程标准
- ▲ 授课计划
- ▲ 程序源代码
- ▲ 习题及答案

高等教育出版社

目录

第 1 章 C 语言概述.....1

1.1 第一个 C 语言程序.....2

1.2 算法与流程图.....3

1.2.1 算法.....3

1.2.2 流程图.....4

1.3 C 语言的发展和语法单位.....5

1.3.1 C 语言的发展与特点.....5

1.3.2 C 语言的主要“单词”.....6

1.3.3 C 语言的主要语法单位.....6

1.4 实现问题求解的过程.....8

1.4.1 问题分析和算法设计.....8

1.4.2 编辑程序.....8

1.4.3 编译程序.....9

1.4.4 运行和调试程序.....9

1.5 技能训练：使用 Visual C++ 6.0

编辑和运行程序.....10

1.5.1 C 程序的开发过程.....10

1.5.2 Visual C++ 6.0 的启动与退出.....11

1.5.3 创建工程和 C 语言程序.....11

1.5.4 编辑、编译和执行 C 语言程序.....13

1.5.5 C 程序文件的管理.....15

本章小结.....16

习题 1.....17

第 2 章 顺序结构.....19

2.1 顺序结构概述.....20

2.1.1 程序解析.....21

2.1.2 常量、变量和数据类型.....21

2.2 基本数据类型.....22

2.2.1 整型.....23

2.2.2 单精度浮点型.....25

2.2.3 双精度浮点型.....26

2.2.4 字符型.....27

2.3 类型转换.....28

2.3.1 自动类型转换.....28

2.3.2 赋值转换.....29

2.3.3 强制类型转换.....29

2.4 数据的输入输出.....29

2.4.1 格式化输出函数 printf().....29

2.4.2 格式化输入函数 scanf().....32

2.4.3 整型数据的输入和输出.....33

2.4.4 浮点型数据的输入和输出.....35

2.4.5 字符型数据的输入和输出.....36

2.4.6 技能训练.....39

2.5 运算符和表达式.....40

2.5.1 算术表达式.....40

2.5.2 赋值表达式.....44

2.5.3 技能训练.....46

2.5.4 关系表达式.....47

2.5.5 逻辑表达式.....48

2.5.6 条件表达式.....52

2.5.7 其他表达式.....53

2.5.8 技能训练.....57

本章小结.....58

习题 2.....59

第 3 章 分支结构.....65

3.1 简单的猜数游戏	66	4.5.2 循环的嵌套结构	106
3.1.1 程序解析	66	4.5.3 技能训练	107
3.1.2 关系运算	67	本章小结	108
3.1.3 单分支、双分支结构	67	习题 4	108
3.1.4 技能训练	70		
3.2 交水费问题	70	第 5 章 函数	121
3.2.1 程序解析	70	5.1 计算圆形面积	122
3.2.2 多分支结构	71	5.1.1 程序解析	122
3.2.3 技能训练	74	5.1.2 函数的定义	123
3.3 算术四则运算	74	5.1.3 函数的调用	124
3.3.1 程序解析	74	5.1.4 函数程序设计举例	126
3.3.2 switch 结构	75	5.1.5 技能训练	127
3.3.3 技能训练	76	5.2 等边三角形问题	128
3.4 考试成绩等级判定问题	77	5.2.1 程序解析	128
3.4.1 程序解析	77	5.2.2 不返回结果的函数	129
3.4.2 多重 if 语句和 switch 结构的比较 ..	79	5.2.3 结构化程序设计思想	130
3.4.3 技能训练	80	5.2.4 技能训练	131
本章小结	81	5.3 年龄问题	134
习题 3	81	5.3.1 程序解析	134
第 4 章 循环结构	89	5.3.2 函数的递归调用	135
4.1 1+2+3+...+100 的计算问题	90	5.3.3 函数程序设计举例	136
4.1.1 程序解析	90	5.3.4 技能训练 (汉诺塔问题)	137
4.1.2 while 语句	90	5.4 两数求较大者问题	139
4.1.3 技能训练	93	5.4.1 程序解析	139
4.2 统计整数的位数	94	5.4.2 局部变量和全局变量	140
4.2.1 程序解析	94	5.4.3 变量生存周期	142
4.2.2 do-while 语句	95	5.4.4 变量存储类型	143
4.2.3 技能训练	96	5.4.5 函数程序设计举例	144
4.3 输出斐波那契数列的前 20 项	97	5.4.6 技能训练	145
4.3.1 程序解析	97	本章小结	146
4.3.2 for 语句	98	习题 5	146
4.3.3 技能训练	100		
4.4 判断素数	101	第 6 章 阶段项目: 小学算术运算考试	
4.4.1 程序解析	101	系统	151
4.4.2 break 语句和 continue 语句	102	6.1 项目需求描述	152
4.4.3 技能训练	104	6.2 系统设计	152
4.5 九九乘法表	105	6.2.1 总体设计	152
4.5.1 程序解析	105	6.2.2 详细设计	153
		6.3 测试与调试	159

本章小结	161
习题 6	162
第 7 章 数组	163
7.1 投票情况统计	164
7.1.1 程序解析	164
7.1.2 一维数组的定义和引用	166
7.1.3 一维数组的初始化	167
7.1.4 技能训练	168
7.2 找出矩阵中最大值所在的位置	174
7.2.1 程序解析	174
7.2.2 二维数组的定义和引用	175
7.2.3 二维数组的初始化	176
7.2.4 技能训练	177
7.3 判断回文	180
7.3.1 程序解析	180
7.3.2 字符数组	181
7.3.3 字符串	182
7.3.4 字符串处理函数	185
7.3.5 技能训练	188
本章小结	191
习题 7	192
第 8 章 指针	199
8.1 找钥匙	200
8.1.1 程序解析	200
8.1.2 地址和指针	201
8.1.3 指针变量的声明	202
8.1.4 指针的基本运算	204
8.1.5 指针变量的初始化	205
8.1.6 技能训练	206
8.2 两值互换	207
8.2.1 程序解析	207
8.2.2 指针作为函数的参数	208
8.2.3 指针程序设计举例	209
8.2.4 技能训练	210
8.3 冒泡排序	211
8.3.1 程序解析	211
8.3.2 <u>指针、数组和地址间的关系</u>	212

8.3.3 指针程序设计举例	214
8.3.4 技能训练	214
8.3.5 数组名作为函数的参数	215
8.3.6 技能训练	216
8.3.7 冒泡排序算法分析	217
8.4 电码加密	217
8.4.1 程序解析	218
8.4.2 字符串和字符指针	219
8.4.3 字符指针程序设计举例	221
8.4.4 技能训练	221
本章小结	222
习题 8	222
第 9 章 结构与共用	227
9.1 建立学生基本信息档案	228
9.1.1 程序解析	228
9.1.2 结构的概念与定义	229
9.1.3 结构变量的定义和初始化	230
9.1.4 结构变量的使用	232
9.1.5 技能训练	233
9.2 计算班级学生平均成绩	234
9.2.1 程序解析	234
9.2.2 结构变量作为函数参数	235
9.3 班级学生成绩排序	236
9.3.1 程序解析	236
9.3.2 结构数组操作	237
9.3.3 技能训练	238
9.4 输出图书基本信息	239
9.4.1 程序解析	239
9.4.2 结构作为函数参数	240
9.4.3 技能训练	242
9.5 共用与枚举	243
9.5.1 程序解析	243
9.5.2 共用类型的定义	244
9.5.3 共用变量的声明与引用	244
9.5.4 枚举类型的定义	246
9.5.5 枚举变量的声明及引用	247
本章小结	249
习题 9	249

第3章 分支结构

【本章工作任务】

- 掌握 if 语句的格式和功能。
- 掌握单分支、双分支、多分支的常用算法。
- 提高使用分支结构解决实际问题的能力。

【本章知识目标】

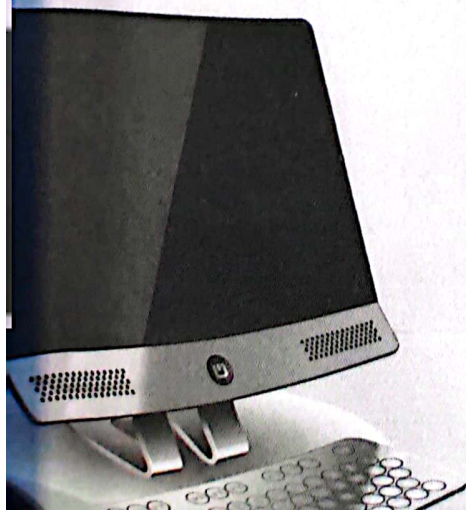
- 掌握分支语句的格式。
- 掌握分支结构的流程。
- 掌握多重分支结构的配对问题。
- 掌握 switch 语句的使用方法。

【本章技能目标】

- 熟练掌握 if 和多重 if 结构的使用。
- 掌握 if 语句中的嵌套关系和匹配原则。

【本章重点难点】

- 多重 if 结构的使用。
- switch 结构的使用。
- 使用多重条件结构实际问题。



通过前面的学习，我们对 C 语言的基本内容（数据表达、运算和流程控制）有初步的认识，能编写一些简单的顺序结构程序。

本章通过典型程序解析，讨论分支结构程序设计的思想和实现。

在执行程序时，一般语句的书写格式顺序执行，但在很多情况下需要根据条件所要执行的语句，这就是分支结构。C 语言中，使用条件语句（if 和 switch）来实现根据条件判断的结果来选择所要执行的程序分支，其中条件可以用表达式来描述，如表达式和逻辑表达式。

3.1 简单的猜数游戏

3.1.1 程序解析

【例 3-1】 简单的猜数游戏。输入读者所猜的整数（假定 1~100 内），与计算机的被猜数比较，若相等，显示猜中；若不等，显示与被猜数的大小关系。

问题分析：

在第 1 章中提到，按照结构化程序设计的观点，任何程序都可以使用 3 种基本控制结构来实现，即顺序结构、分支结构和循环结构。其中分支结构就是根据条件选择执行的语句，一般分为二分支和多分支两种结构。

参考步骤：

- ① 定义一个整型变量 mynumber，同时变量被赋值为 38，即为被猜的数字。
- ② 定义一个整型变量 yournumber，其值提示从键盘输入，即为读者猜的数字。
- ③ 用 yournumber 和 mynumber 比较，分别判别大于、等于、小于的 3 种情况，得出结论。采用多层缩进的书写格式，使程序层次更加分明。

程序代码：

```
/* ex3_1.c */
#include<stdio.h>
void main(){
    int mynumber=38;
    int yournumber;
    printf("请输入您猜的数字:");
    scanf("%d",&yournumber);
    if(yournumber==mynumber)    //判断两数是否相等
        printf("您猜中了!\n");
    else
        if(yournumber>mynumber)
            printf("您猜的数大了!\n");
        else
            printf("您猜的数小了!\n");
}
```

运行程序，输入 3 后的结果如图 3-1 所示。



微课 3-1
分支 例 3-1



注意 >>>>>>>>

条件为一个表达式，此表达式的值可以为真或假，也可以是任意类型的表达式。条件必须括在一对圆括号中。如果条件为真，则执行此语句；如果条件为假，则执行该语句后面的语句。在默认情况下，语句只有1条，如果有多条语句，则需要用大括号“{}”将它们括起来，称为一条“复合语句”或“语句块”，表明在条件为真时，需要同时处理多条语句。

基本 if 语句的控制流程如图 3-3 所示。

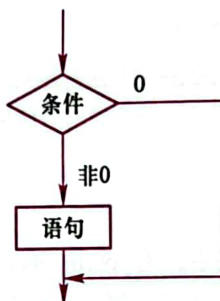


图 3-3 基本 if 语句流程图

【例 3-2】 从键盘输入成绩，判断成绩是否及格，如果及格则输出“您及格了”和“请继续努力!”。

✓ 程序代码:

```

/* ex3_2.c */
#include<stdio.h>
void main(){
    int score;
    printf("请输入您的成绩:");
    scanf("%d",&score);
    if(score>=60){
        printf("您及格了\n");
        printf("请继续努力! \n");
    }
}
  
```

运行结果如图 3-4 所示。

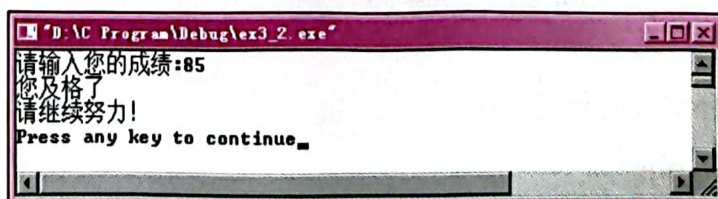


图 3-4 例 3-2 运行结果

程序分析:

本例中声明了变量 score 用于存储成绩值，再利用 scanf() 函数接收用户输入的成绩值。通过 if 语句判断成绩是否大于等于 60 分，如果该关系表达式的值为真，则输出信息：“您及格了”和“请继续努力”，如果该关系表达式的值为假，则程序什么也不显示。请注意满足条件需要做的事情一定要放在一对大括号内，要么都执行，要么都不执行。

2. 双分支结构

除基本 if 语句外, C 语言还提供了 if-else 结构, 其语法形式如下:

```
if(条件)
    语句 1;
else
    语句 2;
```

在这种结构中, 如果条件值为真, 则执行 if 后面的语句 1, 如果条件值为假, 则执行 else 后面的语句 2。

if-else 语句的控制流程如图 3-5 所示。

在了解了 C 语言中 if 结构和 if-else 结构的语法后, 下面来看如何在条件结构中使用关系表达式。

例如, 测试两个数 a 和 b 的大小, 可以写出关系表达式 $a > b$, 然后将其放入 if 语句的条件中, 即 `if(a > b)`。

【例 3-3】 从键盘输入一个数, 判断该数是奇数还是偶数。

问题分析:

判断一个数 a 是奇数还是偶数, 需要判断 a 是否能被 2 整除, 即 a 除以 2 的余数是否等于 0。用 `a%2==0` 来判断表达式 `a%2` 和 0 是否相等实现。

✓ **程序代码:**

```
/* ex3_3.c */
#include<stdio.h>
void main(){
    int a;
    printf("请输入一个数:");
    scanf("%d",&a);
    if(a%2==0)
        printf("%d 是一个偶数\n",a);
    else
        printf("%d 是一个奇数\n",a);
}
```

运行结果如图 3-6 所示。

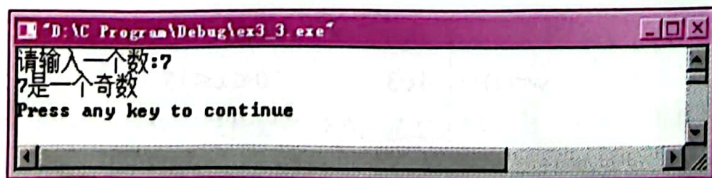


图 3-6 例 3-3 运行结果

程序分析:

程序中调用 `scanf()` 函数读入 a, 再根据事先设定的条件, 去判断 a 除以 2 的余数是否等于 0, 这是典型的二分支结构, 用 if-else 语句实现, 最后调用 `printf()` 函数输出结果。请注意, 双等号 “==” 用于表示相等关系, 而一个等号 “=” 用于表示赋值。

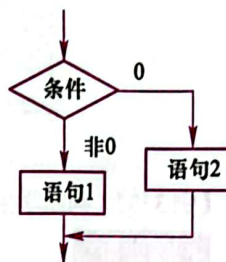


图 3-5 if-else 结构流程图



微课 3-2
分支 例 3-3

【例 3-4】计算个人所得税。根据《个人所得税扣缴申报管理办法（试行）》（国家税务总局公告 2018 年第 61 号）的相关规定，个人所得税免征额为 5000 元/月，个人所得税预扣率表见表 3-3。请输入税前工资（假设小于 8000 元），计算该员工的税后工资。

问题分析：

将用户输入的税前工资和 5000 做比较，有两种可能性。一种征税，一种不征税。程序运行参考结果如图 3-7 所示。

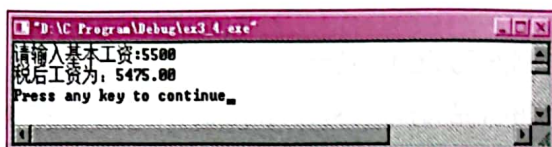


图 3-7 例 3-4 运行结果

【例 3-5】判断学生能否毕业。毕业的条件为体育要达标并且学分要达到 60 分。

问题分析：

学生能否毕业的条件为体育要达标，设变量 sport 为体育是否达标，假定该值等于 y，则体育达标；学分为 score，当 $score \geq 60$ ，学分修够。两个条件之间用 && 运算符连接，表示两个条件都要满足。

程序运行参考结果如图 3-8 所示。

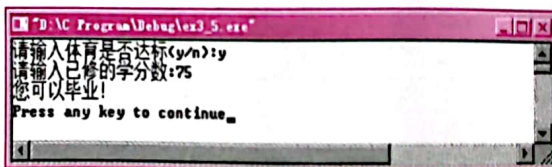


图 3-8 例 3-5 运行结果图

3.2 交水费问题

3.2.1 程序解析

【例 3-6】城市为鼓励市民节约用水，实行根据用水量分段计费。居民应交水费 y （元）与月用水量 x （吨）的函数关系如下，请编程计算（本例为假设的计费方式，实际请参照国家最新标准）。

$$y=f(x)=\begin{cases} 0 & x<0 \\ 4x/3 & 0\leq x\leq 15 \\ 2.5x-10.5 & x>15 \end{cases}$$



微课 3-3

分支 例 3-6

问题分析：

该程序是一个分段函数计算问题。虽然实际生活中月用水量不会小于 0，但程序运行时如果不慎输入一个负数，水费计算将会出错。所以要将各种可能性考虑完整。

程序代码：

```
/* ex3_6.c */
#include<stdio.h>
```



```

✓ void main(){
    double x,y;
    printf("Enter x:");
    scanf("%lf",&x);
    if(x<0)
        y=0;
    else if(x<=15)                //用水量满足  $0 \leq x \leq 15$ 
        y=4*x/3;
    else
        y=2.5*x-10.5;            //用水量满足  $x > 15$ 
    printf("f(%.2f)=%.2fn",x,y);
}

```

运行程序，结果如图 3-9~图 3-11 所示。

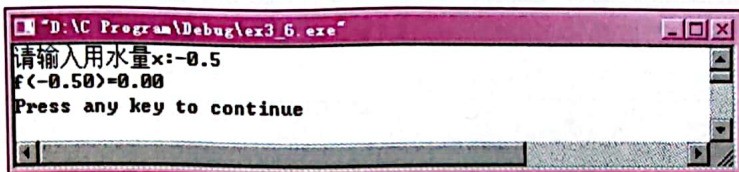


图 3-9 例 3-6 运行结果 1

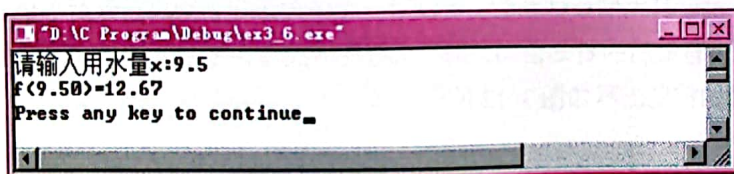


图 3-10 例 3-6 运行结果 2

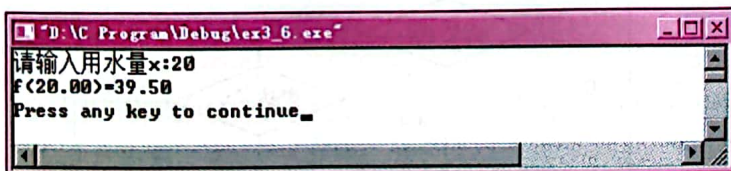


图 3-11 例 3-6 运行结果 3

程序分析:

当 $x < 0$ 时，满足 $y = 0$ 的式子，否则就是 $x \geq 0$ ，在 $x \geq 0$ 的情况下，又分两种情况：一种是 $x \leq 15$ ，综合起来即 $0 \leq x \leq 15$ ；另一种是 $x > 15$ 。

3.2.2 多分支结构

当进行判断时，常遇到这样的情况：如果条件值为真，则执行某些操作；否则，进一步进行条件判断，执行其他操作。例如，外出旅行时选择乘坐火车，那么需要购买火车票；如果火车票卖完了，就乘坐长途汽车；如果长途汽车也没票，那么考虑别的出行方式或者改变出行计划。在程序设计时遇到此类情况，可使用多重 if 结构来解决。

多重 if 结构实际上是 if-else 结构的另一种形式：if-else-if，这种形式也称为阶梯式 if-else-if 结构，这是因为它的外观是阶梯式的。当有多个分支选择时，可采用多重 if 结构。

多重 if 结构的语法如下：

笔记