

全国计算机等级考试（新大纲）

C语言 模拟试题 与模拟 试卷详解

(二级)

朱青 朱华培 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>

-44
/1

7p312-44
ZQ/1

全国计算机等级考试(二级)

C 语言模拟试题与模拟试卷详解

朱 青 朱华培 编著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

050809

内 容 提 要

本书是根据全国计算机等级考试二级(及以上) C 语言考试大纲编写的。

全书主要包括:C 语言程序设计考试指导的要点和技巧,即 C 语言基础知识、控制结构、重点与难点分析;模拟试题详解;模拟试卷和模拟上机操作。

读者对象:全国计算机等级考试 C 语言程序设计应试人员、大专院校师生、相关科技人员及电脑爱好者。

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,翻版必究。

JS362/17

全国计算机等级考试(二级)

书 名: C 语言模拟试题与模拟试卷详解

编 著 者: 朱 青 朱华培

责任编辑: 邓又强

排版制作: 电子工业出版社计算机排版室

印 刷 者:

装 订 者: 北京李史山胶印厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 17.75 字数: 454 千字

版 次: 1998 年 10 月第 1 版 1998 年 10 月第 1 次印刷

书 号: ISBN 7-5053-5019-6
TP·2488

定 价: 22.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。
若书店售缺,请与本社发行部联系调换。电话 68279077

前 言

全国计算机等级考试是国家教委考试中心向全国推出的面向社会的,测试应试者计算机基础理论知识和应用能力的水平考试。C 语言程序设计作为计算机等级考试二级课程之一,同时又是三级 B 及四级操作题所选用的语言,是计算机科学技术的一门专业基础课。

本书依据考试大纲编写了 C 语言程序设计考试指导,从简单的 C 语言基础知识、控制结构的语句语法规则入手,对 C 语言程序设计进行重点与难点的分析。为使读者提高 C 语言的应试能力和全面灵活地掌握 C 语言的程序设计,书中提供了大量的选择题、填空题,并对每题进行了详细的解答与分析。C 语言模拟试卷意在使读者在复习提高的基础上,对自己的水平进行测试。而模拟上机操作篇,则向读者展示了 C 语言在许多方面的应用,是对 C 语言更深一步的学习和研究。

本书注意考试科目的基本知识的归纳总结,在掌握基础知识的同时,以“模拟试题”为主,反复练习,多方面多角度地巩固所学知识。总之,真正掌握这些内容,完全可以达到有关考试的要求。

此书既可作为计算机相关专业学生的参考书,亦可作为计算机爱好者自学的学习资料。同时,又是全国计算机等级考试二级 C 语言程序设计培训班的一本实用教材。

在本书的编写过程中,作者得到了陈德惠老师和阎天纵同志的关心和支持;在出版过程中,得到宗大华教授的帮助。在此一并表示感谢。

由于作者水平所限,加之时间仓促,书中难免存在缺点和疏漏,欢迎广大读者予以指正。

作者

1998.7

目 录

第一篇 C 语言程序设计

一、C 语言程序设计考试指导	(1)
(一) 基础知识	(1)
(二) 控制结构	(7)
(三) 重点与难点	(18)
二、C 语言模拟试题与详解	(29)
(一) 选择题	(29)
(二) 填空题	(98)

第二篇 模拟试卷

一、模拟试卷一	(152)
(一) 选择题	(152)
(二) 填空题	(163)
(三) 模拟试卷一答案	(168)
二、模拟试卷二	(169)
(一) 选择题	(169)
(二) 填空题	(179)
(三) 模拟试卷二答案	(184)

第三篇 模拟上机操作题

一、编写文件加密/解密器	(185)
二、Hanio 塔问题	(188)
三、关键路径(CPM)	(191)
四、飞机自动订退票系统	(196)
五、迷宫问题	(204)
六、统计图形的编制	(211)
七、求最短路径	(215)
八、计算机辅助考核	(222)
九、骑士周游问题	(237)
十、全屏幕文本编辑器	(248)

第一篇 C 语言程序设计

一、C 语言程序设计考试指导

(一) 基础知识

C 语言是一种具有效率高、可移植性好、书写简结灵活、数据类型丰富的结构化程序设计语言。它是处于汇编语言和高级语言之间的一种过渡型程序语言, C 语言既适合用来编写系统软件, 又适合用来编写应用软件。因此, C 语言是目前国际上广泛流行的, 并很有发展前景的计算机程序设计语言。

1. 基本程序结构

C 语言是由函数构成的。一个 C 源程序至少包含一个 main 函数, 也可以包含一个 main 函数和若干个其它函数。可见 C 程序的基本单位是函数。

每个函数定义有如下形式:

类型 函数名(参数表)

参数说明

{ 数据说明部分

 执行语句部分

}

其中: 类型部分确定函数的类型, 即限定了函数返回值的类型, 如 float, char * 等。函数名是一个标识符。参数表是由一系列以逗号隔开的变量名组成, 而参数说明则是指出函数参数表中变量的类型。

数据说明部分和执行语句部分组成函数体, 它是实现程序功能的具体表现形式。为了增加程序的易读性, 往往要加入注释语句。

程序举例: 在计算机的终端上输入一段正文, 统计其中的行、单词和字符的数量。

```
/* count the number of lines, words and chars */  
#include <stdio.h>  
main()  
{  
  int nl,nw,nc,inword;  
  char ch;  
  nl=nw=nc=0;  
  inword=0;  
  while((ch=getchar())!=EOF)
```

```

{ nc++;
  if (ch == '\n') nl++;
  if (ch == ' ' || ch == '\t' || ch == '\n')
    inword = 0;
  else if (inword == 0)
  { inword = 1;
    nw++;
  }
}
printf("line = %d word = %d character = %d", nl, nw, nc);
}

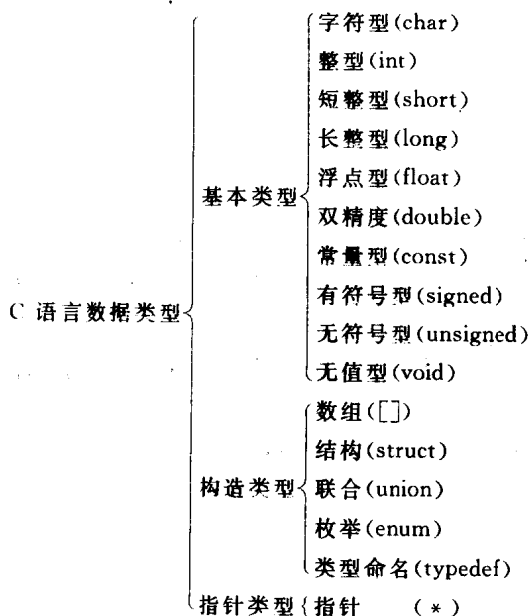
```

此程序是由一个 main 函数实现的,程序前部带有 # 号的语句是编译预处理语句。由 /* 和 */ 表示注释语句,在 Borland C++ 中也用 // 表示注释,注释不能嵌套,而且编译时不产生目标代码。

2. 基本语法

(1) **标识符** 实际上是一个字符序列。在 C 语言中通常用标识符标记变量、常量、数据类型、函数和程序的名字。合法的标识符是:以字母或下划线符中任一字符开头;后面可以是任意的字母、下划线符和数字组成的字符序列。如: _out2, Sum, page_2 都是合法的标识符,而 return \$5, 7example 是不合法的。

(2) **数据类型** C 语言提供的数据类型十分丰富。可以将其分为基本类型、构造类型和指针类型。表示如下:



(3) **关键字** 也称保留字,在 C 语言中赋予它们特定含义的单词。通常不能用作一般标识符。这些关键字有: char, long, short, int, unsigned, float, double, enum, union, struct, void, sizeof, typedef 是用于数据类型的; auto, extern, register, static 是用于存储类; break, case, continue, default, do, else, for, goto, if, return, switch, while 是用于语句的。还有一些特定字: define, include, undef, ifdef, endif 和 line 是用于 C 语言的预处理中,一般也不用作用户标识符,以免造成混乱。

(4) 常量与变量

常量 在程序运行过程中,其值不能被改变的量称为常量。常量包括:整常量,即十进制、八进制和十六进制以及长型整常量,浮点常量,字符常量和字符串常量。

- 十进制整常量:如 123、-32767、0。
- 八进制整常量:以 0 开头的数表示,如 0123,即 $(123)_8$ 。
- 十六进制整常量:以 0x 开头的数表示。如 0x123,即 $(123)_{16}$, -0x18 即 $-(18)_{16}$ 。
- 长型整常量:在一个整常量后面加一个字母 l 或 L,则认为是 long int 型常量。
- 浮点常量:有两种表示形式。十进制数形式:由数字和小数点组成,如:0.123、123.、0.0。指数形式:如 123E4 代表 123×10^4 ,字母 e(或 E)之前必须有数字,且 e 后面指数必须为整数。
- 字符常量:是用单引号(即撇号)括起来的一个字符,如 'a', '\$'。此外,还允许出现转义字符,就是以 "\" 开头的字符序列。例如,经常在 printf 函数中出现的 '\n',它代表一个'换行'符。'\ddd' 表现 1 到 3 位八进制所代表的字符。
- 字符串常量:是一对双引号括起来的字符序列。如:"Beijing", "\$123.45",都是字符串常量。注意:'d' 是字符常量,而"d"是字符串常量。

变量 顾名思义,在程序运行过程中,其值变化的量称为变量。对用户而言,变量代表某个具体的数字、字符等。对系统而言,变量则是指计算机内存中的存储空间,用于存放各种数据。C 语言中,变量分为整型变量、实型变量、双精度实型变量、字符型变量、数组变量、指针变量、结构变量等几种数据类型。变量在使用之前,必须加以定义说明。

- 整型变量的说明形式:

int <变量 1>, <变量 2>, ..., <变量 n>;

在 Turbo C2.0 中规定整型变量取值范围为 -32768~+32767。

- 实型与双精度实型变量的说明格式:

float <变量 1>, <变量 2>, ..., <变量 n>;

double <变量 1>, <变量 2>, ..., <变量 n>;

实型 float 的取值范围为 $10^{-38} \sim 10^{38}$;双精度型 double 的取值范围为 $10^{-308} \sim 10^{308}$

- 字符变量的说明形式:

char <变量 1>, <变量 2>, ..., <变量 n>;

字符型 char 取值包括数字字符、英文字母、标点符号,即 ASCII 代码表中的字符。故在 ASCII 代码表中变量值是一 128~127 之间的所有字符。

(5) 表达式 C 语言中的表达式类似于数学中的表达式,它是由常量、变量、运算符、括号、函数等有机地连结而成的式子。表达式分成:数学表达式、逻辑表达式、赋值表达式和逗号表达式。当然,表达式与运算符是不可分的。

C 语言的运算符有以下几类:

- 算术运算符 (+、-、*、/、%、++、--)
- 关系运算符 (>、<、==、>=、<=、!=)
- 逻辑运算符 (!、&&、||)
- 位运算符 (<<、>>、~、|、^、&)
- 赋值运算符 (=、+=、-=、*=、/=、%=、>>=、<<=、&=、^=、|=)
- 条件运算符 (? :)

- 逗号运算符 (,)
- 指针运算符 (* 和 &)
- 求字节数运算符 (sizeof)
- 类型转换运算符 (类型)
- 结构体成员运算符 (* →)
- 下标运算符 ([])

算术表达式 是由算术运算符、类型转换运算符和运算码组成。

运算符	含义	结合方向	优先级
++ -- (类型)	自增 自减 类型转换	从右至左	高  低
* / %	乘法 除法 求余	从左至右	
+ -	加法 减法	从左至右	

例如：合法的 C 算术表达式

$a/b+c-1.5+'a'$ 实型算术表达式。
 (double) (389 % 'c' / y) 实型算术表达式。
 ++i 整型算术表达式。

[题目] 求下面算术表达式的值。

(1) $x+y\%3*(\text{int})(x+z)\%2/4$

设 $x=2.5, y=7, z=4.7$

(2) $++i-j++$ 设 $i=5, j=7$;

[解答] (1) 将 x, y, z 代入表达式。

$$2.5+7\%3*(\text{int})(2.5+4.7)\%2/4=2.5$$

$$(2) ++i-j++=(++i)-(j++)=6-7=-1$$

说明： $++i, --i$ 是在使用 i 之前，先使 i 的值加(减)1。而 $i++, i--$ 是在使用 i 之后，使 i 的值加(减)1。

关系表达式 是将两个值进行比较，判断比较的结果是否符合给定的条件，其结果或是“真”，或是“假”。关系运算符的优先级低于算术运算符。

运算符	含义	结合方向	优先级
<	小于	自左向左	高 ↓ 低
<=	小于等于		
>	大于		
>=	大于等于		
==	等于	自左向右	
!=	不等于		

例如：合法的关系表达式

$$(a+b)>c, \quad x\%y+z!=0$$

〔题目〕 求下面关系表达式的值。

$$a+b>a+c \quad \text{设 } a=1, b=2, c=3$$

〔解答〕 将 a、b、c 代入表达式 $1+2>1+3$ ，显然其值是假，即 0。

逻辑表达式 将关系表达式和逻辑符连接起来构成逻辑表达式。逻辑运算符的优先级高于关系运行符。

运算符	含义	结合方向	优先级
!	逻辑非	自左向右	高 ↓ 低
&&	逻辑与		
	逻辑或		

例如：合法的逻辑表达式

$$(a+b>c)\&\&(a+c>b)\&\&(b+c>a)$$

〔题目〕 求下面逻辑表达式的值：

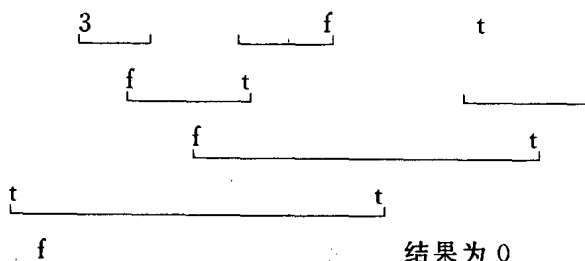
$$(1) \ 6\&\&0 \ || \ 2\&\&5$$

$$(2) \ b<(i+j<3)\&\&! (i==j) \ || \ (i!=3)\&\&b$$

设 $i=2, j=1, b$ 为真值 't'

〔解答〕 (1) 先求 && 后作 || 结果为 1。

$$(2) \ b<(i+j<3) \ \&\& \ ! \ (i==j) \ || \ (i!=3) \ \&\& \ b$$



字位表达式 用位运算符连接的整型量，结果为整数。字位表达式也可以认为是整型表达式的一种特殊形式。

运算符	含义	结合方向	优先级
~	取反	自右向左	高 ↓ 低
<< >>	左移 右移	自左向右	
& ^	按位与 按位或 按位异或	自左向左	

例如：合法的位表达式

$b \gg (m-n+1), \quad (c \ll n) \& (b \gg m)$

[题目] 求下面位表达式的值。

(1) $\sim (15 \ll 4)$

(2) $060 | 017$

[解答] (1) 15 的二进制表示 0000000000001111

$15 \ll 4:$ 0000000011110000

$\sim (15 \ll 4):$ 1111111100001111

若用八进制表示结果为 0177417。

(2) $060 | 017$ 是将八进制数 60 与八进制数 17 进行按位或运算。

即 $00110000 | 00001111 = 00111111$

若用八进制表示结果为 077。

赋值表达式 将赋值号“=”右侧表达式的值赋给赋值号左边的变量。在赋值号“=”之前加上其它运算符,可以构成复合的运算符。赋值表达式的一般形式为:

<变量> <赋值运算符> <表达式>

运算符	含义	结合方向
= += -= *=	赋值运算符	自右至左
/= %= >>= <<=		
&= ^= =		

例如:合法的 C 赋值表达式

$a = 5 + b * c - d / 4 * \cos(x)$

$x \% = y + 3$

[题目] 求下面赋值表达式的值。

(1) $y = (a + b) * c - \sin(x),$

设 $a = 3, b = 2, c = 2, x = 3.1415926/6$

(2) $b * = b + 8,$ 设 $b = 2$

[解答] (1) 将 a, b, c, x 值代入

$$y = (3+2) * 2 - \sin(\pi/6) = 10 - 0.5 = 9.5$$

(2) 将 $b=2$ 代入:

$$b * = b + 8, \text{ 相当于 } b = b * (b + 8) = 2 * (2 + 8) = 20.$$

条件表达式 由条件运算符($?:$)和表达式组成。其表示形式是:

逻辑表达式? 表达式 1: 表达式 2

求解过程: 逻辑表达式的值若为非零(真)时, 则条件表达式的值等于表达式 1 的值; 若逻辑表达式的值为零(假), 则条件表达式的值等于表达式 2 的值。

例如: $(i == 0) ? a + b : a - b$

$(x > y) ? x : y$

[题目] 写出下列 C 语言的条件表达式。

(1) 将大写字母 ch 转换成小写字母的表达。

(2) 有三个整数 a, b, c , 表示其中最大的数。

[解答] (1) $(ch >= 'A' \&\& ch <= 'Z') ? (ch + 32) : ch$

(2) $(a > b ? a : b) > c ? (a > b ? a : b) : c$

首先求出 a 与 b 的最大值 $(a > b ? a : b)$, 再将其与 c 构成条件表达式。

逗号表达式 用逗号将两个或多个表达式连接起来。逗号表达式的一般形式为:

表达式 1, 表达式 2, …… , 表达式 n

逗号表达式的求解过程是: 先求表达式 1, 再求表达式 2, …… , 最后求表达式 n 。整个逗号表达式的值是表达式 n 的值。例如:

$$7 + 8, 3 + 4$$

$$(x = 3.1 * 2, x / 2), x + 6$$

[题目] 分别求下列式子中 x, y, z, w 的值。

(1) $w = (x = 1, y = 2, z = 3, 4)$

(2) $w = x = 1, y = 2, z = 3, 4$

[解答] (1) $x = 1, y = 2, z = 4, w = 4$

(2) $x = 1, y = 2, z = 4, w = 1$

(二) 控制结构

C 语言提供了丰富的语句。按结构化程序设计方法, 可将程序的控制结构分为: 顺序结构、选择结构、循环结构。即结构化程序由若干个基本结构组成, 每一个基本结构可以包含一个或若干个语句。

1. 简单语句

简单语句有表达式语句、空语句和复合语句, 属于顺序程序结构。

表达式语句 由一个表达式构成一个语句。常见的表达式语句有: 赋值语句和函数调用语句。简而言之, 表达式语句就是由表达式后跟随一个分号而成的。

赋值语句 形式为:

标识符=表达式;

功能:将赋值号“=”右边表达式的值赋给左边的标识符。例如: $i=i+1$; $z=(x>0)?x:0$;

$Bool=(y \% 4==0) \&\& (y \% 100 != 0) \parallel (y \% 400 == 0)$;都是赋值语句。

函数调用语句 由一次函数调用后跟随一个分号构成函数调用语句。这一类语句十分丰富,根据 C 语言不同版本中的库函数,可以形成各种功能的函数调用语句,但要用预编译命令 `#include` 说明。

putchar 函数 字符输出函数,即向终端输出一个字符。例如:

putchar(c);

输出字符变量 c 的值。

printf 函数 格式输出函数,一般表示形式为:

printf(格式控制,输出表列);

其中:“格式控制”是用双引号括起的字符串,它含有普通字符和格式说明(即:由“%”和格式字符组成,如 `%d`, `%c`, `%f` 等)。“输出表列”需要输出的数据或表达式。

例如: `printf("a=%d b=%4.1f",a,b);`

`printf("string=%s char=%c a=%x",str,ch,a);`

都是合法的输出函数表达式。

[题目] 写出下面程序的输出结果。

```
#include <stdio.h>
main()
{
    int a=18;
    float b=123.45;
    char ch=' $ ';
    char *s="abcd";
    printf("a1=%d \t",a)
    printf("a2=%x \t",a)
    printf("a3=%o \n",a)
    printf("b=%6.1f \t",b)
    printf("ch=%c s=%s \n",ch,s);
}
```

[解答] 按照输出格式:`%d` 输出整型十进制,`%x` 输出十六进制,`%o` 输出八进制;`%f` 输出浮点数,`%6.1f` 表示此浮点数占 6 位,其中小数部分占 1 位;`%c` 输出一个字符,`%s` 输出字符串。所以上面程序的输出结果是:

`a1=18 Ha2=0X12 a3=022`

`b=123.4 ch= $ s=abcd`

getchar 函数 字符输入函数,一般表示形式为:

c=getchar();

函数的值就是从输入设备得到的字符。例如:`c=getchar()`;若键盘输入' \$ ',则 c 的值是' \$ '。

scanf 函数 格式输入函数,一般表示形式为:

`scanf(格式控制,地址表列);`

其中:“格式控制”的含义同 printf 函数,“地址表列”是由若干个地址组成的表列,可以是变量的地址,或字符串的首地址。

例如: `scanf("%d,%c,%2.1f",&i,&ch,&x);`

`scanf("%d,%d,%d",&a,&b,&c);`

都是合法的输入函数表达式。

[题目] 当输入流为“12345 067891a36”时,写出下面程序的输出结果。

```
#include <stdio.h>
main()
{ int i,j;
  float x,y;
  char c;
  scanf("%2d%f%f%c%d\n",&i,&x,&y,&c,&j);
  printf("%2d,%5.1f,%7.1f,%c,%d",i,x,y,c,j);
}
```

[解答] 程序的输出结果是:

12, 345.0, 67891.0, a, 36

空语句 只有一个分号的语句,说明它什么也不做,空语句的表示形式为:

`;`

空语句有时用来作被转向点,或循环语句中的空循环体。特别要注意的是如果在编写程序时,分号“;”的位置放错了,可能将某个语句变成空语句了。例如下面程序段:

```
for(i=1;i<=10;i++);
printf("%d\t",i);
```

程序本意为输出 1 到 10 的 10 个数字,而在 for() 后加了一个分号,使循环体变成了空语句,即输出 11 了。

复合语句 可以用{ }把一些语句括起来成为复合语句。实际上复合语句就是一个分程序结构。例如,{ i++; sum+=i; res=y;}是一个复合语句。复合语句在语法上等价于一个简单语句。在循环语句中常常以复合语句作为循环体。

2. 条件语句(if)

条件语句的一般形式为:

```
if (表达式)
    语句 1
else
    语句 2
```

求解过程是:首先求出表达式的值并以此为条件,当条件为真,则执行语句 1;若条件为假,则执行语句 2。在 C 语言中,以表达式的值为 0 表示假,非 0 表示真。此语句是选择结构。

注意以下几点:

- ① 语句 1 和语句 2 都可以是复合语句。
- ② else 语句 2 可以缺省,使语句成为:
if (表达式) 语句 1;
- ③ 条件语句是可以嵌套的。为避免语句执行的二义性,C 语言规定:else 是与同一分程序的前面最接近的,不带 else 的 if 相对应。例如:

```
if (i>=1)
    if (j<=2)
        result=5;
    else result=6;
```

则 else 与前面的第二个 if 相对应。如果需要 else 与第一个 if 对应,则必须用花括号{}改写为:

```
if (>=1)
{
    if (j<=2)
        result=5;
}
else result=6;
```

[题目] 选择题 若 k 是 int 型变量,且有下面的程序片段:

```
k=-3;
if (k<=0) printf("####");
else printf("&&&&");
```

上面程序片段的输出结果是:

- (A) #### (B) &&&&
(C) ####&&&& (D) 有语法错误,无输出结果。

[解答] 答案为(A)。值得注意的是:如果 else 之前的 printf() 后没有分号“;”,则此题目有语法错。

[题目] 写出下面程序的输出结果。

```
main ()
{
    int a=1,b=5,c=2;
    int x,y;
    scanf("%d",&x);
    if(x<0) y=a+b*x*x;
    else if (x==0) y=c;
    else y=c-b*x*x;
    printf("y=%d",y);
}
```

若 x 的输入数据是 3。

[解答] x=3,则由 if 语句可知 $y=c-b*x*x$; 故 $y=2-5*3*3=-43$ 。

3. 开关语句(switch)

开关语句的一般形式为：

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

求解过程是：先计算 switch 后面的表达式，当其结果与某一个 case 后面的常量表达式的值相等时，就执行此 case 后面的语句，若所有 case 中的常量表达式的值都未能与该表达式的值匹配，就执行 default 后面的语句。此语句是选择结构。

注意以下几点：

- ① 在同一个开关语句中，每个 case 的常量表达式的值必须互不相同。
- ② default 部分可以缺省。若开关语句中不包含 default，并且不存在与 case 后面常量表达式相匹配的情况，则不执行任何动作。
- ③ 通常在使用开关语句时，希望只执行某个 case 常量表达式 i 或 default 后面的某个语句 $i(1 \leq i \leq n+1)$ ，并不想执行所有 $j > i$ 的语句 j，因此，可在每个 case 的语句 i 后面，都加上 break 语句。一旦某个 case 被匹配，其后的语句被执行，遇到 break 则执行完此 switch 语句。

例如： switch(ch)

```
{
    case '<': b=x<y;
    case '>': b=x>y;
    case '=': b=x=y;
    default: printf("Error \n");
}
```

是合法的开关语句

[题目] 求下面程序的输出结果：

```
main()
{
    int score;
    scanf("%d \n", &score);
    switch(score % 10)
    {
        case 10:
        case 9: printf("A \n"); break;
        case 8:
        case 7: printf("B \n"); break;
        case 6: printf("C \n"); break;
        default: printf("NO _pass \n");
    }
```

若输入数据是 85。

[解答] 结果是 B。此程序将学生分数 score 转成四级分别为 A、B、C、No_pass, A(90—100), B(70—89), C(60—69), NO_pass(低于 60)。

[题目] 求下面程序的输出结果。

```
#include <stdio.h>
main ()
{ int x=1,y=0,a=0,b=0;
  switch(x)
  {case 1:
    switch(y)
    { case 0: a++;break;
      case 1: b++;break;
    }
    case 2:
      a++; b++; break;
  }
  printf("a=%d, b=%d\n",a,b);
}
```

上面程序的输出结果是:

- (A) a=2,b=1 (B) a=1,b=1
(C) a=1,b=0 (D) a=2,b=2

[解答] 正确答案是 (A)。由初始化可知: x=1; y=0。

语句执行 switch(x)

```
case 1:
  switch(y)
  case 0: a++; break;
```

又知: a=0, b=0; 此处 a=1, b=0; 而 break 语句只退出 switch(y), 在 switch(x) 的 case 1 后没有 break; 故仍需执行 switch(x) 的 case 2 后的语句 a++; b++; 所以 a=2, b=1。

4. 循环语句

C 语言的循环语句共有三种: 即 while, for 和 do_while。显然属于循环控制结构。

while 语句 用来实现“当”的循环结构。其一般形式如下:

while (表达式)
语句

求解过程: 表达式是判断循环能否继续下去的条件, 而语句是循环体。当表达式为非 0 值时, 执行 while 语句中的循环体, 否则退出循环。

注意以下几点:

- ① 循环体如果包含一个以上的语句, 应用花括弧括起来, 以复合语句形式出现。