

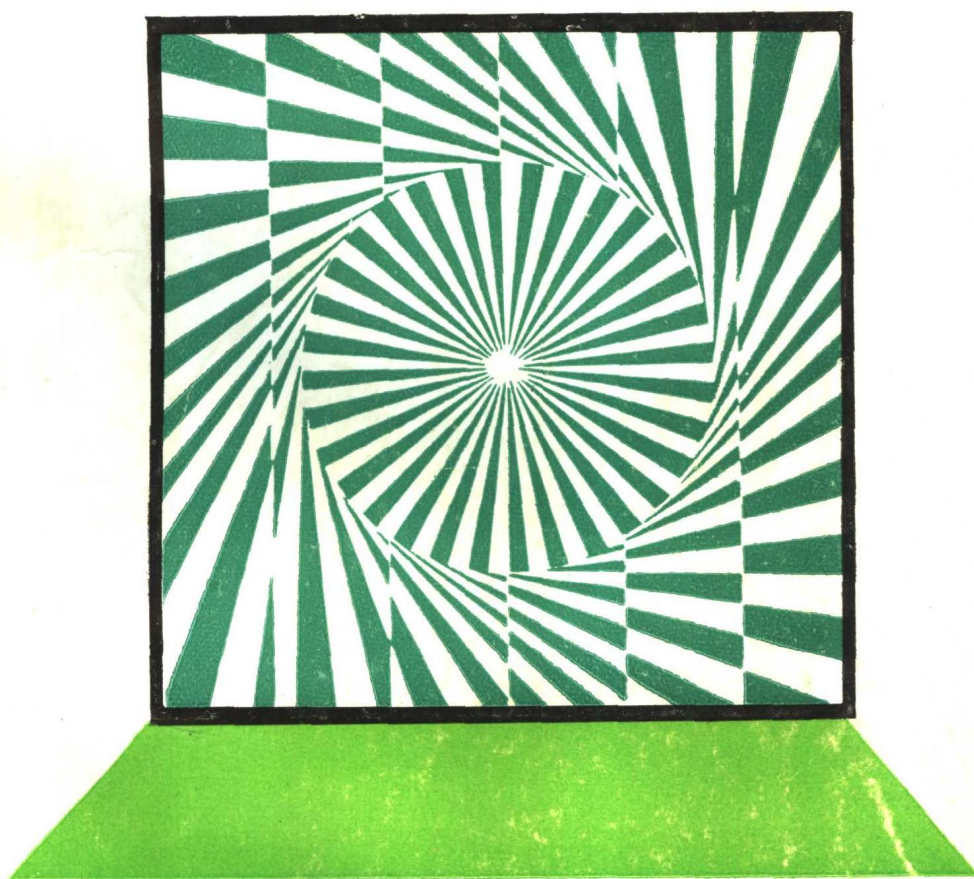
92

全国计算机应用软件人员水平考试辅导丛书

C 语言学习指导

(初级程序员级 程序员级)

李文兵 编著



清华大学出版社

12
4

全国计算机应用软件人员水平考试辅导丛书

C 语言学习指导

(初级程序员级 程序员级)

李文兵 编著

清 华 大 学 出 版 社

内 容 简 介

本书为计算机应用软件人员水平考试辅导丛书之一，是在北京软件人员水平考试委员会实施办公室的指导下，由中国软件行业协会考试指导中心具体组织编写的。

本书包括六章：C语言语法特点、基本内容、主要问题与重要例题、上机指导、C语言考试纲要和试题分析。

本书可供初级程序员级及程序员级计算机应用人员学习、参考。可与《IBM PC C语言简明教程》、《IBM PC C语言 例题 习题 库函数》配合使用，作为大专院校配套教材。对于参加计算机专业自学考试的人员也非常适用。

C 语言学习指导

李文兵 编著

*

清华大学出版社出版

北京 清华园

北京北方印刷厂印刷

新华书店总店科技发行所发行

开本：787×1092 1/16 印张：4.25 字数：102千字

1990年7月第1版 1990年7月第1次印刷

印数：0001~4000

ISBN7-302-00787-X

定价：2.75元

出 版 说 明

计算机应用软件人员水平考试是造就宏大的多层次计算机应用人才队伍的一项重要措施，几年来的实践证明，它对激励在职计算机应用人员提高软件水平，配合国家对软件人员技术职称评定和职务聘任以及加强对外软件人才交流等方面都起了积极的作用。

为了帮助各类应用软件人员准备应试，并供各地举办辅导班参考，我们邀请了北京大学、清华大学、中国科学院研究生院等许多高校的计算机专家，组成编委会，陆续编辑出版《全国计算机应用软件人员水平考试辅导丛书》。

本丛书包括各个级别的辅导教材及题例解答。李文兵根据《计算机应用软件人员水平考试大纲》的要求编写的《C语言学习指导》一书，包括了C语言课程的基本内容、上机指导与试题分析，可供初级程序员级及程序员级计算机应用人员学习、参考。

欢迎国内专家与广大读者对本丛书提出批评意见。

中国软件行业协会考试指导中心

1990.6

前 言

C语言既是成功的系统描述语言,又是通用的程序设计语言,是目前国际上非常流行的计算机语言,也是目前国外软件开发的主流语言。我国从90年代起正式将C语言列为计算机应用软件人员水平考试语种。为适应这种形势,在中国软件行业协会考试指导中心的指导下,编写了这本书。

本书所介绍的六章内容是广大软件工作者和学员所关心的问题,也是学员应当掌握的。

只有了解C语言的语法特点,才能有的放矢地去掌握C语言。它的基本内容应是学员学习的主要内容,也是学员应掌握的范围。书中第三章的主要问题是从小语言的基本语法和应用中归纳出来的,共有15个问题,其中前10个是针对《IBM PC C语言简明教程》一书各章写的,建议初级程序员和程序员掌握;而后五个问题只要求程序员掌握。重要例题是为学员彻底弄清每个问题所举的典型程序。无论哪门计算机语言,只有经过上机实际操作,才能体会深刻,C语言也不例外。因此,希望广大学员都要想方设法,按上机指导,切实练习一番。这样,才会有很大收获。考试纲要是作者根据教学大纲提出来的,供学员参考。最后一部分是试题分析,是想让学员了解试卷上可能出现的题型及其分析办法。

愿本书能对大家有所帮助。预祝大家能尽快掌握C语言。

徐国平副教授审阅了文稿。在编写出版过程中,还得到了北京科海培训中心的热情支持与赞助,在此向他们表示衷心的感谢。

由于水平有限,时间仓促,错误和不妥之处,诚恳希望广大读者与专家指正。

李文兵

一九九〇年六月十三日

目 录

第一章	C语言语法特点.....	(1)
第二章	基本内容	(4)
第三章	主要问题和重要例题	(19)
第四章	上机指导	(43)
第五章	C语言考试纲要	(52)
第六章	试题分析	(55)

第一章 C语言语法特点

要学习C语言，就必须了解C语言的特点，也只有了解了C语言的特点，才能有的放矢地去掌握C语言，学习C语言。那么，C语言有哪些特点呢？本章就介绍这个问题。

1. 以小写字母为基本书写形式

C语言是以小写字母作为其基本书写文字的，这主要是为适应人们的习惯。为此，就连关键字等也都是用小写字母书写的。我们在编写C语言源程序时，一般用小写字母编写。只有宏定义所定义的宏名，为了使其与一般变量相区别，习惯上使用大写字母。大家应注意，凡是用小写字母所定义的变量名、函数名以及各种标识符，在使用时，还得用小写字母，这是因为C语言是把大写的和小写的分别作为两个名字处理的缘故。

2. 数据类型多

数据类型较多，也是C语言的一个特点。在C语言里，作为基本数据类型的有：char型，int型，short型，long型，unsigned型，float型，double型；作为复合类型（导出）的有：指针型，数组，结构，共体。此外，有返回值的函数也可看作是一种复合型数据类型，其返回值必须是算术型和指针型。在现代C语言里，还有枚举（enum）类型。

3. 运算符多

C语言比其它语言有较多的运算符。

作为算术运算符有：+（加）、-（减）、*（乘）、/（除）和%（取余）。

作为逻辑运算符有：&&（逻辑与）、||（逻辑或）和!（逻辑非）。

作为位逻辑运算符有：&（位积）、|（位和）、^（位差）、<<（左位移）和>>（右位移）。

作为赋值运算符有简单赋值运算符“=”和10个复合赋值运算符：+=、-=、*=、/=、%=、<<=、>>=、&=、^=和|=。

作为关系运算符有：>（大于）、<（小于）、>=（大于或等于）、<=（小于或等于）、==（相等）、!=（不等）。

作为递增递减运算符有：++（递增）、--（递减）。

作为指针运算符有：*（取值）、&（取址）。

此外，还有：?:（条件运算符）、,（逗号运算符）、（类型名）（强制类型转换运算符）、sizeof（求字节数运算符）、-（负值运算符）。

4. 适于结构化程序的控制语句

(1) if 语句

(2) if 语句1, else 语句2

(3) while (e) 语句

(4) do语句, while (e)

(5) for (e1; e2; e3) 语句

(6) break

(7) continue

(8) switch (e)

```
{ case c1;
    语句 1;
    break;
  case c2;
    语句 2;
    break;
    :
  default;
    语句n + 1;
    break;
}
```

(9) goto

5. 程序由函数组成

C语言源程序是由一个或多个相当于其它语言的子程序的函数组成的，特点如下：

(1) 每个可执行的C源程序都要有一个主函数main()，它是可执行的函数。不管它放在什么位置，C程序就是从它开始执行，到其返回结束。

(2) 其他函数要由主函数main()或另外函数调用才能执行。

(3) 这些函数可以放在一个文件里，也可分别放在几个文件里。

(4) 函数在定义时不允许嵌套，但可嵌套调用。

(5) 调用顺序与定义顺序无关。

6. 表示方法简洁

C语言的表示方法比任何语言都要简洁；就是与其姐妹语言PASCAL相比，也要简洁得多，这在《IBM PC C语言简明教程》中已有介绍。最典型的如下：

(1) { 和 } 相当PASCAL或ALGOL语言中的begin和end

(2) $i = i + 2$ 可以写成

$i += 2;$

(3) $i = i + 1$ 可以写成

$i++$ 或 $++i;$

(4) 允许使用

$i = j = k = 0;$

(5) 函数可以出现在条件表达式中，如

$\text{if } ((c = \text{getchar}()) == 'g');$

(6) 关键字用得较少，基本的有28个。现代C语言又多了几个，请参看《IBM PC C语言简明教程》第5页。

7. 可用指针指向各类数据

C语言的指针可指向基本数据、数组、函数和结构，指针还可以指向指针。在C语言中，指针是非常重要的，恰当地使用它可编写出结构紧凑简洁的程序，但是指针又是较难理解的

部分。因此，可以这么说，谁掌握了指针，谁才掌握了C语言。

8. 语法中没有定义输入输出功率

在C语言里，输入输出功能是以库函数的形式提供的，而不是包含在语法里。例如，`getchar( )`就是如此，它是从标准输入设备输入一个字符的函数。源程序经编译程序编译后，再经连接程序链接，就把该函数的功能附加到可执行程序上。

第二章 基本内容

C语言的基本内容，即我们应掌握的范围是：数据类型、运算符、语句、宏预处理、库函数、程序结构。这里，在介绍这些内容的同时，并把重点提示给大家，以使大家掌握重点，带动一般。

1. 数据类型

(1) 基本数据类型及其表示范围如表2.1所示。

表2.1 基本数据类型

数 据 类 型	所 占 位 数	数 值 范 围
char	8	- 128 ~ + 127
unsigned char	8	0 ~ 255
short	16	- 32768 ~ + 32767
unsigned short	16	0 ~ 65535
int	16	- 32768 ~ + 32767
unsigned int	16	0 ~ 65535
long	32	- 2147483648 ~ + 2147483647
unsigned long	32	0 ~ 4294967295
float	32	- 1.0e38 ~ + 1.0e38
double	64	- 1.0e308 ~ + 1.0e308

(2) 非图形控制字符的扩展用法 非图形控制字符及一些字符常量的扩展用法如表2.2所示。

表2.2 非图形字符

非图形字符	扩 展 用 法	功 能
NL (LF)	\n	回车换行
HT	\t	水平制表
BS	\b	退 格
CR	\r	回 车
FF	\f	走 纸
\	\\	反 斜 杠
,	\,	单 引 号
"	\"	双 引 号
ddd	\ddd	三位 8 进制数

说明：

①非图形控制字符是指ASCII编码值在 0X00~0X1F之间的字符。

②\ddd用法中的ddd，可以是一位，二位，至多为三位。特例为\0，其ASCII编码值

为 0。

(3) 复合数据类型 是指指针型、数组型、结构型、共体型、函数型和枚举型,共6种。

所谓指针类型就是指指向某数据类型的指针,其类型就看该指针是指向什么样的数据类型。凡是指向同一数据类型的指针都可作为同一类型使用。

数组类型是某一种数据类型的数据的有序排列,序号是从 0 开始,如下所示:

int	int	int	int	int
0	1	2	3	4

该图表示该数组为整型数组,其每个元素都必须是整型数据。这就是说,数组的数据类型是由其元素的数据类型决定的。

结构(struct)类型是由几种类型的数据(除函数类型外)所组合而成的结构数据。显然,数组只是结构数据的一种特殊情况。以short类型、char类型和float类型为结构元素的结构如下所示:

short	char	float
-------	------	-------

结构的元素称作成员。

共体(union)是具有所指定的数据类型(除函数类型外)之中一种数据类型值的结构数据,其大小取其元素中最大的一种数据类型的大小。

枚举(enum)其数据结构定义形式为

enum 枚举类型名 { 枚举表 } 变量表

其中枚举类型名和变量表是任选项。enum型数据的例子如:

enum color { red, green, blue };

enum color tv;

这里,color是枚举类型名,tv为该枚举类型的一个变量。

这时,若打印变量tv的值,即

printf(“%d %d %d”, red, green, blue);

则输出为 0 1 2。可见,枚举型数据为整型常量。其枚举表中的各标识符不仅能按 0、1、2……取值,而且能设定为任何整型常量。例如,对于上例来说,若按如下初始化:

enum color { red=100, green, blue };

则red、green、blue就分别对应100、101、102。

枚举型变量用到控制结构中很方便。

例1 用于for结构中

```
main( )
{ enum color tv;
  for (tv = red, tv <= blue, tv ++ )
  { ...
  }
}
```

例2 用于switch结构中

```
switch (tv)
{ case red; 语句1; break;
  case green; 语句2; break;
  case blue; 语句3; break;
  default; 语句4;
}
```

巧妙地运用enum数据类型,可使程序易读易懂。

(4) 存储类别 在C语言里,在说明一个变量时,除了说明其数据类型外,还可说明其存储类别,以决定变量在存储器中的存储位置,以及生存周期。

存储类别有自动、静态、外部和寄存器四种,分别用关键字auto、static、extern和register来表示。自动存储变量仅在其所定义的模块中有效,进入到该模块其值存在,出了该模块其值丢失。静态存储变量也仅在其所定义的模块中有效,但其值在整个执行过程中都是存在的。外部存储变量在程序执行的全过程中都是存在的,无论哪部分程序都可使用。寄存器存储变量是把寄存器作为存储区域,以提高存取数据速度。注意,有些C编译程序,如优化(86)是把register存储类别作为auto类别处理的。若想访问IBM PC机的中央处理器的寄存器,定义一个如下所示的结构变量regs即可。

```
struct{ int ax, bx, cx, dx, si, di, ds, es; }regs;
```

结构中的各个成员都是IBM PC机中央处理器的寄存器。

(5) 变量的初始化 变量可按如下形式进行初始化,即:

```
int i = 10;
char c = 'a';
```

静态存储变量和外部静态存储变量在没有初始化时,其值为0;但自动存储变量和寄存器存储变量却不然。

数组在模块内进行初始化时,必须说明为静态存储类别,如:

```
static int a[5] = { 0, 1, 2, 3 };
```

这时,数组a的初始值为0, 1, 2, 3, 0, 即未给初始值的元素a[4]的值为0。

说明语句:

```
static a[ ] = { 1, 2, 3, 4, 5, 6 };
```

使数组a的大小为给出的数据的个数,此时为6。

结构可按如下形式初始化:

```
struct {
    int i;
    char c;
    } a = { 5, 'a' };
```

(6) typedef

关键字 typedef 可用来定义数据类型,如:

```
typedef int, I, *Ip;
I j, k;
```

lp p, q;

分别定义了整型变量j、k和指向整型数据的指针p、q。

(7) 类型转换

基本类型可按如下规则进行转换:

①char或short型在运算时要转换为int型;进行浮点运算时, float型要转换为double型。

②若有一个操作数是double型, 则另一个操作数也要转换为double型, 且运算结果也是double型。

③如果操作数中没有double型, 就看有没有long型, 若有一个操作数为long型, 则另一个操作数也要转换为long型, 且运算结果也是long型。

④如果操作数中既没有double型, 又没有long型, 就看有没有unsigned型, 若有一个操作数为unsigned型, 则把另一个操作数也转换为unsigned型, 运算结果也是unsigned型。

⑤如果双方都是int型, 则运算结果也是int型。

(8) 字段结构 所谓字段结构是指结构成员是不满一个字的整型数据的结构数据, 其成员要按如下定义:

unsigned a: 1;

其中a是字段名, 必须用关键字unsigned来说明, 1表示该字段所占的位数是1位。

遇到位数为0的字段, 其下面的字段要移到下一个字单元去存。无名的字段, 其给出的位数全置空。在IBM PC机上, 字段分配方向是从一个字的低位开始的。

2. 运算符

学习C语言的运算符, 除了要掌握每种运算符的功能外, 还要清楚每种运算符的优先级和结合性。C语言共有44个运算符, 分15个优先级。每种优先级别的结合性如下表所示。只有2、13、14三类为右结合性, 其余都是左结合性。

优先级	运 算 符	结 合 性
1	() [] · →	右结合
2	* & - ! ~ ++ -- (类型) sizeof	
3	* / %	
4	+ -	
5	<< >>	
6	< > <= >=	
7	= = ! =	
8	&	
9	^	
10		
11	&&	
12		
13	? :	右结合
14	= + = - = * = / = % = · > = < = & = ^ = ! =	右结合
15	,	

下面按优先级顺序介绍各类运算符的功能。

(1) () [] · → 这些运算符的优先级别最高, 如:

a * (b + c)
month [9]
birth · year
birthp → day

最后一个也可写成

(* birthp) · day
a · b · c 等价于
(a · b) · c

(2) 单项运算符

① *p 表示p为指针, 该式可用来访问p所指向的目标。

② &d 表示取变量d的地址, 该式可用来给出指向d的指针。

③ -a 表示取a的负值, unsigned类型的负值为 2ⁿ 与该数之差。在C语言里, 没有正号 (+)

④ !a 对a进行逻辑非运算: 若a为 0, 则结果为整数 1; 若a不为 0, 则结果为整数 0。

⑤ ~a 对a按位取反, a必须是整型数据。

⑥ ++i, --i 和 i++, i-- 这四种运算都是使i的值加 1 或使i的值减 1, 但运算符放在i前和i后是有区别的, 如:

i = 3;
j = ++i;

的运算结果, j的值为 4; 而

i = 3;
j = i++;

的运算结果, j的值仍为 3。

⑦ (type)a 这是强制把a的数据类型转换为type类型, 如函数sin()要求其参数必须是double型。若i为int型, 这时函数sin()就必须按如下形式书写:

sin((double) i)

⑧ sizeof a 和 sizeof (type) 是分别求变量a和type数据类型 的字节个数。

(3) 乘法运算符 这里指如下三种运算符:

表达式 * 表达式
表达式 / 表达式
表达式 % 表达式

注意, 在第三种运算中, 操作数不能是float型。

(4) 加法运算符 包括 + 和 -, 其运算表达式如下:

表达式 + 表达式
表达式 - 表达式

上述两个表达式是分别求和或差。指针也可以使用 + 和 -, 加减一个整数, 但必须注意, 指

针所加减的整数要乘上一个比例因子，以使指针在移动时，总能指向某数据。显然，比例因子就是指针所指数据类型的字节个数。指向同一数据类型的指针也可进行减运算。

(5) 移位运算符 包括 $\ll$ 和 $\gg$ ，可构成如下表达式：

表达式 $\ll$ 表达式

表达式 $\gg$ 表达式

注意，表达式的值必须是整型数据。当右边表达式的值为负数或比左边表达式的值的位数还大时，结果没有意义。

对于右位移来说，若左边表达式的值为unsigned型，则为逻辑移位，否则，为算术移位。逻辑右移时，左边空出的位为0；算术右移时，左边空出的位要补上符号位的值。

(6) 关系运算符 有六种，可构成如下运算：

表达式 $<$ 表达式 (小于时为真)

表达式 $>$ 表达式 (大于时为真)

表达式 $\leq$ 表达式 (小于或等于时为真)

表达式 $\geq$ 表达式 (大于或等于时为真)

表达式 $=$ 表达式 (相等时为真)

表达式 $!=$ 表达式 (不等时为真)

这些运算分别表示小于、大于、以下和以上。若成立，则值为1；若不成立，则值为0。指向同一数组的指针也可进行这种运算。

这些运算符是左结合性，因此，下式：

$i = 4;$

$l = 8 > i > 3$

就相等于

$l = (8 > i) > 3$

故l值为0；而

$8 > i \&\& i > 3$

则相当于

$(8 > i) \&\& (i > 3)$

故l值为1。

(8) 位积运算符 由它所构成的运算表达式如下所示：

表达式 $\&$ 表达式

表达式的值必须是整型数据，其运算是按位进行逻辑与。

(9) 位差运算符 由它所构成的运算表达式如下所示：

表达式 $\wedge$ 表达式

其中表达式的值必须是整型数据，其运算是按位求逻辑异或。

(10) 位和运算符 由它所构成的运算表达式如下所示：

表达式 $|$ 表达式

其中表达式的值必须是整型数据，其运算是按位求逻辑或。

(11) $\&\&$ 运算符 它是逻辑与运算符。当其两边的操作数都不是0时，运算结果为1；否则为0。

该运算符是左结合性。一旦出现值为 0 的操作数，就无须再对右边的操作数进行运算，如

`if(i == 0 && ++j == 2)`

若 `i` 为 0，则左操作数的值为 1，然后进行 `j` 值 + 1，同 2 比较；但若 `i` 值不为 0，左操作数的值就是 0，整个运算的结果就是 0，也就无须再进行右操作数的运算。

(12) `||` 运算符 它是逻辑或运算符。两个操作数中只要有一个其值不为 0，运算结果就是 1；而当两个操作数的值都是 0 时，运算结果就是 0。

该运算符也是左结合性。一旦出现值不为 0 的操作数，就无须再对其右边的操作数进行运算，如在条件表达式

`a == 'a' || i++ == 0 || --k == 5`

中，设 `a` 值为 6，`i` 值为 8，`k` 值为 6，仅到判断 `i` 值止，`i` 值变为 9，`k` 值不递减仍为 6。

(13) 条件运算符 其用法如下所示：

表达式 1 ? 表达式 2 : 表达式 3

运算过程是先判别表达式 1，若其值不为 0，则整个表达式的结果为表达式 2 的解；否则，为表达式 3 的解。如表达式

`max = a > b ? a : b;`

是把 `a` 和 `b` 中大者的值赋值给变量 `max`。注意，表达式 2 和表达式 3 不允许用不同类型的指针指定。

(14) 赋值运算符 赋值运算符分为简单赋值运算符和复合赋值运算符。简单赋值运算符的用法是

`e = 表达式`

其功能是把表达式的值赋值给 `e` 所指定的存储单元。复合赋值运算符的用法是

`e1 op = e2;`

`op` 代表 +、-、*、/、%、<<、>>、&、|、^，其功能是

`e1 = e1 op e2;`

(15) 逗号运算符 它的用法是

表达式 1, 表达式 2

具有左结合性，其结果是右边表达式的值。该运算符常用在 `for` 语句中，用来对循环控制变量进行初始化和修改。

3. 语句

C 语言的语句和其他语言一样，一般是顺序执行。控制语句能改变程序流向。这里，介绍 C 语言里的各种语句。

(1) 表达式语句 表达式的后边加上分号就是 C 语言的一个语句，如：

`i = j * 6;`

`c = getchar ( );`

等都是表达式语句。在 PASCAL 语言里，分号是作为语句的分隔符使用的，而在 C 语言里，分号是用来表示语句的终止。因此，模块里最后一个语句也必须加分号。

(2) 复合语句 其形式如下：

{ 说明语句序列

执行语句序列

}

可以一部分缺省或两部分都缺省。

说明语句中所说明的变量，仅在该复合语句中有效，该语句开始执行时，进行初始化。

复合语句用在必须用两个以上C语句，而用一个语句不能表达的时候。这时，只要用{和}把这些语句围起来，即可构成复合语句。

(3) if语句 它有三种形式，即：

①if (表达式) 语句

②if (表达式) 语句1 else语句2

在这两种形式中，无论哪种都是首先要判别通常叫做条件表达式的表达式，若其值不为0，则执行语句1；否则，在①中便什么也不执行，而在②中执行语句2。

当if语句发生嵌套时，else同最近的if配对，如：

if (表达式) if (表达式) 语句 else语句

被解释为

if (表达式) { if (表达式) 语句 else语句 }

若写成

if (表达式) { if (表达式) 语句 } else语句

else则与第1个if配对。

③if (e1) 语句1

else if (e2) 语句2

else if (e3) 语句3

.....

else 语句n+1

这是if语句构成多分支结构的一种用法，哪个条件成立，就执行其对应的语句；若e1~en都不成立，则执行语句n+1。

(4) while语句 其用法如下：

while (表达式) 语句

只要表达式的值不为0，就连续执行语句。表达式的判别要在每次语句执行前进行。下面的程序，其执行结果为：s为55，i为0。

int i=10, s=0;

while (i>0)

{ s+=i; i--; }

(5) do语句 其用法如下：

do 语句 while (表达式);

语句要连续执行到表达式的值为0止，同while语句的区别是，表达式的判别要在每次语句执行后进行。

(6) for语句 其用法如下：

for (表达式1; 表达式2; 表达式3) 语句

其功能等效于：