

项 目 四

▶ 计算个人所得税



学习目标

- 了解浮点型相关知识
- 掌握结构体的含义
- 掌握结构体数组的定义和使用
- 掌握引入文件的意义
- 具有文件基本操作的能力



学习路径





任务一 计算个人所得税后输出



浮点型

(1) 浮点型概念

浮点型用于描述现实生活中的实数，例如 1.2、123.45 等，基本类型为 `float`。可以根据取值 的范围和数据精度的不同，通常情况下，将浮点数分为单精度（`float`）和双精度（`double`）。

类型	说明	字节数	有效数字
<code>float</code>	单精度浮点型, 取值范围: $-3.4E+38 \sim 3.4E+38$	4	6~7
<code>double</code>	双精度浮点型, 取值范围: $-1.7E+308 \sim 1.7E+308$	8	15~16



浮点型

(2) 浮点型常量

浮点型也称为实型。浮点型常量也称为实数或者浮点数。在 C 语言中，实数只采用十进制。它有二种形式：十进制小数形式和指数形式。

1) 十进制数形式：由数码 0~9 和小数点组成。

1.0、15.8、5.678、-0.13、500.、-267.8230 等均为合法的实数。

2) 指数形式：由十进制数，加阶码标志“e”或“E”以及阶码（只能为整数，可以带符号）组成。其一般格式为： aE_n （ a 为十进制数， n 为十进制整数，表示阶码），其值为 $a \times 10^n$ 。

3.2E5（等于 3.2×10^5 ），4.7E-2（等于 4.7×10^{-2} ），0.6E7（等于 0.6×10^7 ）



(3) 浮点型变量

C 语言中浮点型变量指的就是实数变量（存放可以带小数的数据的变量），分为两类：**float**（单精度型）和 **double**（双精度型）。

```
float d;      /* 声明浮点型变量 d*/  
double real; /* 声明双精度型变量 real*/
```



格式输出函数 printf()

(1) 其中类型符号用以表示输出数据的类型

C 语言中浮点型变量指的就是实数变量（存放可以带小数的数据的变量），分为两类：**float**（单精度型）和 **double**（双精度型）。

格式字符	意义
d	以十进制形式输出带符号整数(正数不输出符号)，如果是长整型数据前面一个加上字符“l”。
o	以八进制形式输出无符号整数(不输出前缀 0)。
x,X	以十六进制形式输出无符号整数(不输出前缀 0x)。
u	以十进制形式输出无符号整数。
f	以小数形式输出单、双精度实数，如果不指定输入宽度，整数部分全部输出，输出 6 位小数(可能不是有效数据)。
e,E	以指数形式输出单、双精度实数。
g,G	以 %f 或 %e 中较短的输出宽度输出单、双精度实数。
c	输出单个字符。
s	输出字符串。



格式输出函数 printf()

(2) 精度：精度格式符以“.”开头，后跟十进制整数。本项的意义是：如果输出数字，则表示小数的位数；如果输出的是字符，则表示输出字符的个数；若实际位数大于所定义的精度数，则截去超过的部分。



表达式中的类型转换

(1) 自动转换

在某种范围内整型数据可以和字符型数据通用，而整型是浮点型的一种特殊形式。因此，整型、浮点型和字符型数据之间可混合运算。例如： $3.45+10+'a'-2.5*'c'$ ，是合法的。在混合运算时，编译系统先将不同数据类型数据自动转换成同一类型，再进行运算。



自动转换遵循以下规则：

- ①若参与运算量的类型不同，则先转换成同一类型，然后进行运算。
- ②转换按数据长度增加的方向进行，以保证精度不降低。
- ③所有的浮点运算都是以双精度进行的，即使仅含 `float` 单精度量运算的表达式，也要先转换成 `double` 型，再作运算。
- ④ `char` 型和 `short` 型参与运算时，必须先转换成 `int` 型。
- ⑤在赋值运算中，赋值号两边量的数据类型不同时，赋值号右边量的类型将转换为左边量的类型。



(2) 强制类型转换

除了自动类型转换之外，程序设计人员还可以根据运算的要求，在程序中强行将数据的类型进行转换，称为强制类型转换。强制类型转换是通过类型转换运算来实现的，其一般格式如下：

(类型说明符) (表达式)



格式说明

- ①强制类型转换符的功能是把表达式的运算结果强制转换成类型说明符所表示的类型。
- ②类型说明符和表达式都必须加括号（单个变量可以不加括号）。
- ③无论是强制转换或是自动转换，都只是为了本次运算的需要而对变量的数据长度进行 的临时性转换，而不改变数据说明时对该变量定义的类型



if-else 嵌套结构

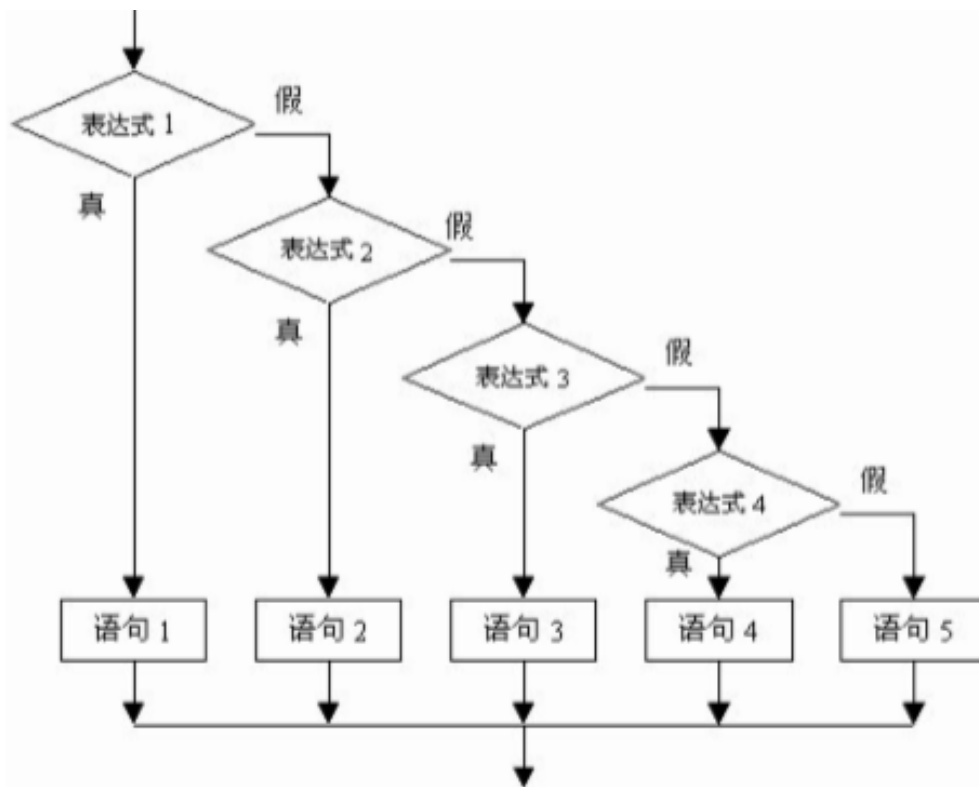
在程序设计中，经常使用级联的 if-else-if 实现多路分支结构一般格式：

```
if ( 表达式 1)  
    语句 1;  
else if ( 表达式 2)  
    语句 2;  
else if ( 表达式 3)  
    语句 3;  
...  
else if ( 表达式 m)  
    语句 m;  
Else  
    语句 m+1
```



if-else 嵌套结构

在程序设计中，经常使用级联的 if-else-if 实现多路分支结构一般格式：





条件语句使用小结

一个表达式的返回值都可以用来判断真假，除非没有任何返回值的 `void` 型和返回无法判断真假的结构。当表达式的值不等于 0 时，它就是“真”，否则就是假。一个表达式可以包含其他表达式和运算符，并且基于整个表达式的运算结果可以得到一个真或假的条件值。因此，当一个表达式在程序中被用于检验其真或假的值时，就称为一个条件。



(1) if 语句

if (表达式) 语句 1; 如果表达式的值为非 0，则执行语句 1，否则跳过语句继续执行下面的语句。如果语句 1 有多于一条语句要执行时，必须使用 { 和 } 把这些语句包括在其中，此时条件语句形式为：

```
if(表达式)
{
语句 1;
}
```



(2) if-else 语句

除了可以指定在条件为真时执行某些语句外，还可以在条件为假时执行另外一段代码。在 C 语句中利用 else 语句来达到这个目的。if—else 语句格式为：

```
if( 表达式 ) 语句 1;  
else 语句 2
```



(3) if-else 嵌套

条件语句可以嵌套，这种情况经常碰到，但条件嵌套语句容易出错，其原因主要是不知道哪个 if 对应哪个 else。其语句格式为：

```
if( 表达式 1)
语句 1;
  else if( 表达式 2)
语句 2;
  else if( 表达式 3)
语句 3;
  .....
else
语句 n
```



(4) switch-case 语句

执行 `switch` 开关语句时，将变量逐个与 `case` 后的常量进行比较，若与其中一个相等，则执行该常量下的语句，若不与任何一个常量相等，则执行 `default` 后面的语句。其语句格式为：

```
switch( 变量 )  
{ case 常量 1:  
    语句 1 或空; break;  
  case 常量 2:  
    语句 2 或空; break;  
  .....  
  case 常量 n:  
    语句 n 或空; break;  
  default:  
    语句 n+1 或空;  
}
```



任务二 将税率表存放在结构体数组中，
然后再计算 个人所得税并输出



结构体的意义

首先考虑解决下面一个问题，输入一个学生的学号、姓名、性别、班级以及他的数学、外语和语文的三科课程成绩，求出他的总分和平均分。

问题分析：通过该项目的问题描述，可知通过顺序结构就可以编写出这个项目的解决程序，用四个字符数组变量分别表示学号、姓名、性别、班级；用三个浮点类型变量表示学生的三科成绩，然后通过计算求出总分和平均分并输出即可。



结构体类型的定义

C语言提供了一个重要的构造数据类型——结构体类型，来解决复杂事物表示问题，它将多个数据项集合到一个数据类型中，每个数据项目被称为数据成员，它们可以是不同的数据类型，既可以是基本数据类型，也可以是另一种构造数据类型。结构体数据类型的一般定义如下：

```
struct 结构体名  
{  
    数据类型 1 成员名 1;  
    数据类型 2 成员名 2;  
    数据类型 3 成员名 3;  
    .....  
    .....  
    数据类型 n 成员名 n;  
};
```



结构体类型变量的定义

上面声明了一个结构体类型，它相当于一个模型，其中并无具体数据，只有定义了具有结构体类型的变量之后，才能在其中存放具体的数据。在 C 语言中，可以使用三种方式定义结构体类型的变量：

(1) 直接声明结构体类型变量

这种方式要求在 `struct` 后不使用结构体名，例如：

```
struct{  
.....  
} st1, st[2];
```




结构体类型变量的定义

(2) 先定义结构体类型，再单独声明结构体类型变量

这种方式要求先定义结构体类型，再由一条单独的语句声明变量，例如：

```
struct student{  
.....  
};  
struct student st1, st[2];
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/568135053020006120>