



思维导图

选择结构程序设计

案例引入——成绩管理中成绩的判断处理

任务描述

假设学生成绩表如表 4-1 所示。

表 4-1 学生成绩表

no	name	t1	t2	t3	t4	avg	sick	l_e	absent	f_e	score	grade
1	刘延运	80	60	80	60	70				96.0	85.6	
2	孙燕	95	95	90	80	90	1			86.0	85.6	
3	王珊珊	95	98	90	80	90.8				95.0	93.3	
4	刘斌	90	95	95	99	94.8				75.5	83.2	
5	王一山	96	95	99	99	97.2				64.5	77.6	
6	胡志民	98	90	90	85	90.8				91.5	91.2	
7	张瑜	98	95	99	90	95.5				89.5	91.9	
8	李冰	95	95	90	80	90	1			55.0	67.0	
9	张珊	85	99	99	99	95.5				89.5	91.9	
10	陈明明	95	90	95	90	92.5				90.0	91.0	

(1) 试编写程序,根据终结性考核成绩(score)给出相应考核的水平等级。

(2) 试编写程序,找出终结性考核成绩(score)的最高分和最低分。

说明: D 不及格(<60),C 及格($60\sim79$),B 良好($80\sim89$),A 优秀($90\sim100$)。

思考: 如果要对很多学生的成绩进行以上处理有什么更好的方法。请预习本书的下一模块,在其中找到办法。



任务目标

- (1) 能够用逻辑表达式描述客观条件。
- (2) 能够应用分支结构设计算法。
- (3) 培养软件开发必备的逻辑思维能力。



源代码展示

- (1) 根据终结性考核成绩(score)给出相应考核的水平等级。

```
#include "stdio.h"
main()
{
    float score;
    char ch;
    scanf("%f",&score);
    switch((int)(score/10.0))
    {
        case 10:
        case 9:  ch='A';break;
        case 8:  ch='B';break;
        case 7:
        case 6:  ch='C';break;
        default: ch='D';
    }
    printf("score= %.1f,grade= %c\n",score,ch);
}
```

- (2) 找出终结性考核成绩(score)的最高分和最低分(这里假设 3 个学生的成绩,10 个学生成绩的编程思路以此类推)。

```
#include <stdio.h>
main()
{
    float s1,s2,s3,min,max;
    printf("please input s1,s2,s3:");
    scanf("%f%f%f",&s1,&s2,&s3);
    min=max=s1;
    if(s2<min)min=s2;
    else if(s2>max) max=s2;
    if(s3<min)min=s3;
```

```

else if(s3>max) max=s3;
printf("min= %.1f,max= %.1f\n",min,max);
}

```



运行结果

(1) 根据终结性考核成绩(score)给出相应考核的水平等级。
模拟运行结果如图 4-1 所示。

```

c:\ "D:\TC\实训\Debug\shixun4.exe"
85.6
score=85.6,grade=B
Press any key to continue

```

图 4-1 模拟运行结果 1

(2) 找出终结性考核成绩(score)的最高分和最低分。
模拟运行结果如图 4-2 所示。

```

c:\ "D:\TC\实训\Debug\shixun4.2.exe"
please input s1,s2,s3:85.6 93.3 77.6
min=77.6,max=93.3
Press any key to continue_

```

图 4-2 模拟运行结果 2



程序分析

程序(1): 成绩等级有 4 种, 所以要判断成绩所在的范围, 采用多分支语句。由于成绩有规律的等级划分, 所以用 switch 比较方便, 注意表达式的使用(int)(score/10.0)。

程序(2): 从一组数据中找最大值(max)、最小值, 我们先设置最大值(max)、最小值(min)都为第一个数, 这样后面的数据依次与当前的最大值、最小值比较就可以了。最后 max、min 中存储的就是最大值和最小值。



知识讲解

C 语言提供了可以进行逻辑判断的若干选择语句, 由这些选择语句构成程序中的选择结构, 又称为分支结构。选择结构是结构化程序设计的 3 种基本结构之一, 在大多数结构化程序设计中都会遇到选择问题, 因此熟练运用选择结构进行程序设计是程序员必须具备的能力。本模块将详细介绍如何在 C 程序中实现选择结构。

4.1 关系运算符和关系表达式

关系运算是逻辑运算中比较简单的一种,关系运算就是比较运算,即将两个值进行比较,判断是否符合或满足给定的条件。如果符合或满足给定的条件,则称关系运算的结果为“真”;如果不符合或不满足给定的条件,则称关系运算的结果为“假”。

4.1.1 关系运算符

C语言提供了6种关系运算符: $<$ (小于)、 $>$ (大于)、 \leq (小于等于)、 \geq (大于等于)、 $=$ (等于)、 $!=$ (不等于)

注意:由两个字符组成的运算符之间不允许有空格,如“ \leq ”不能写成“ $<=$ ”。

关系运算符的优先级如下。

(1) $<$ (小于)、 $>$ (大于)、 \leq (小于等于)、 \geq (大于等于)关系运算符的优先级相同, $=$ (等于)、 $!=$ (不等于)也相同。前4种运算符的优先级高于后两种。

(2)关系运算符的优先级低于算术运算符。

(3)关系运算符的优先级高于赋值运算符。

关系运算符是双目运算符,具有自左向右的结合性。

【例 4-1】 关系运算符优先级的实例。

$c > a + b$ 等价于 $c > (a + b)$	关系运算符的优先级低于算术运算符。
$a > b = c$ 等价于 $(a > b) = c$	“ $>$ ”优先级高于“ $=$ ”。
$a = b < c$ 等价于 $a = (b < c)$	“ $<$ ”优先级高于“ $=$ ”。
$a = b > c$ 等价于 $a = (b > c)$	关系运算符的优先级高于赋值运算符。

若 $a=5, b=3, c=1$, 则 $a > b > c$ 的计算过程是:先计算 $a > b$, 结果为真, 值为 1; 再计算 $1 > c$, 结果为假, 值为 0, 所以 $a > b > c$ 表达式为假。

4.1.2 关系表达式

用关系运算符将两个表达式(算术、关系、逻辑、赋值表达式等)连接起来所构成的表达式称为关系表达式。

关系表达式的值是一个逻辑值,即“真”或“假”。C语言没有逻辑型数据,以1代表真,以0代表假。

【例 4-2】 关系表达式的实例。

假如 $a=3, b=2, c=1$, 则关系表达式 $a > b$ 的值为真, 即表达式的值为 1; 关系表达式 $b + c < a$ 的值为假, 即表达式的值为 0。

又例如, $5 = 3$ 值为 0, $5 > 3$ 值为 1, 则 $a > b > c$ 值为 0。

描述各种简单条件的应用如下。

儿童: $\text{age} \leq 6$

老人: $\text{age} \geq 60$



测试
等级考试真题

及格: $\text{score} \geq 60$
 判断整数 n 为偶数: $n \% 2 == 0$

4.2 逻辑运算符和逻辑表达式

用逻辑运算符将关系表达式或逻辑量连接起来的式子就是逻辑表达式。下面介绍 C 语言中的逻辑运算符和逻辑表达式。

4.2.1 逻辑运算符

C 语言提供了以下 3 种逻辑运算符。

- (1) ! 逻辑非: 条件为真, 运算后为假; 条件为假, 运算后为真。
 - (2) && 逻辑与: 相当于日常生活中的“而且”、“并且”, 只在两条件同时成立时为真。
 - (3) || 逻辑或: 相当于日常生活中的“或”, 两个条件只要有一个成立即为真。
- 其中, && 和 || 是双目运算符, ! 为单目运算符。

4.2.2 逻辑表达式

! 表达式

表达式 1 && 表达式 2

表达式 1 || 表达式 2

表 4-2 为逻辑运算的真值表。用它表示当 a 和 b 的值为不同组合时, 各种逻辑运算所得到的值。

表 4-2 逻辑运算的真值表

a	b	! a	! b	a && b	a b
1	1	0	0	1	1
1	0	0	1	0	1
0	1	1	0	0	1
0	0	1	1	0	0



测试
等级考试真题

在一个逻辑表达式中如果包含多个逻辑运算符, 则按照以下的优先级进行计算。

- (1) ! (逻辑非) 的优先级最高, && (逻辑与) 次之, || (逻辑或) 最低。
- (2) && 和 || 的优先级低于关系运算符, ! 高于算术运算符。
- (3) 逻辑运算符的优先级高于赋值运算符。

【例 4-3】 逻辑运算符优先级的实例。

$a > b \&\& x > y$ 等价于 $(a > b) \&\& (x > y)$

$a == b || x == y$ 等价于 $(a == b) || (x == y)$

$! a || a > b$ 等价于 $(! a) || (a > b)$

【例 4-4】 非 0 值作为逻辑值参与运算。

非 0 值作为逻辑值参与运算=真时与 1 的作用一样。

若 $a=4$, 则 $!a=0$ (假)。

若 $a=4, b=5$, 则 $a \& b=1$ (真), $a || b=1$ (真), $!a || b=1$ (真)。

$4 \& 0 || 2=1$ (真), c (真) $\& \& d$ (真) $=1$ 。

从例子还可以看出,系统给出的逻辑运算结果不是 0 就是 1,不可能是其他数值。而在逻辑表达式中作为参与逻辑运算的运算对象可以是 0 (作为假)也可以是任何非 0 的数值(按真对待)。事实上,逻辑运算符两侧的对象不但可以是 0 和 1 或者是 0 和非 0 的整数,也可以在任何类型的数据(如字符型、实型、指针型)。

需要注意以下两点。

(1)在 C 程序中,不能出现“连不等式”。在数学知识里,用形如 $0 < x < 100$ 的不等式表示 x 的取值范围是正确的,但在 C 语言中,若写成 $0 < x < 100$ 则是错误的,当 x 的值为负数时,先判断 $0 < x$,不成立用 0 表示,再用 0 值与 100 比较,自然 $0 < 100$ 成立,结果为 1。所以上式在 C 语言中应表达成 $0 < x \& \& x < 100$ 。

(2)在逻辑表达式的求解中,并不是所有的逻辑运算符都被执行,只是在必须执行下一个逻辑运算符才能求出表达式的解时,才执行该运算符。

例如, $a \& \& b \& \& c$, 只有 a 为真,才需要判别 b 的值;只有 a, b 都为真,才需要判别 c 的值;只要 a 为假,整个表达式已经确定为假,就不必判别 b, c ;如果 a 为真, b 为假,不判断 c 。

$a || b || c$, 只要 a 为真,整个表达式已经确定为真,就不必判断 b 和 c ;只有 a 为假,才判断 b ; a, b 都为假才判断 c 。

【例 4-5】 如果 a, b, c, d, m, n 分别为 1、2、3、4、1、1,分析整个表达式 $(m=a > b) \& \& (n=c > d)$ 的结果和 m, n 的值。

由于 $a > b$ 为假(0),所以赋值后 $m=0$,赋值表达式 $m=a > b$ 也为 0。此时整个表达式的结果已经知道(0),所以不进行表达式 $n=c > d$ 的计算,表达式计算结束后, $n=1$ (未改变)。

掌握 C 语言的关系运算符和逻辑运算符后,可以用一个逻辑表达式来表示一个复杂的条件。

例如,判断儿童和老人: $age \leq 6 || age \geq 60$ 。

例如,要判断某一年 $year$ 是否是闰年。闰年的条件是符合下面二者之一:能被 4 整除,但不能被 100 整除;能被 4 整除,又能被 400 整除。

上面两个条件可以用一个逻辑表达式来表示。

```
(year % 4 == 0 && year % 100 != 0) || year % 400 == 0
```

可以用一个 $!$ 来判别非闰年。

```
!( (year % 4 == 0 && year % 100 != 0) || year % 400 == 0)
```

也可以用以下逻辑表达式判别非闰年。

```
(year % 4 != 0) || (year % 400 == 0 && year % 100 == 0)
```

4.3 if 语句

if 语句用来判定所给定的条件是否满足,根据判定的结果(真或假)决定执行给出的两种操作之一。

4.3.1 简单 if 语句

简单 if 语句的语法格式如下。

```
if(表达式) 语句
```

功能是判断表达式是否成立,若成立,则执行语句,不成立,则执行下一条语句,如图 4-3 所示。

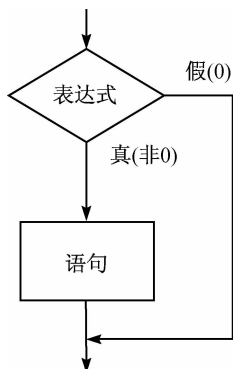


图 4-3 if 语句流程图

例如,统计及格人数。

```
if(score >=60)
{
    printf("及格");
    c1=c1+1;
}
```

【例 4-6】 从键盘输入两个不相等的数,存入 a 和 b,判断 a 和 b 的大小,操作实现使 a 的值大于 b 的值。

程序代码如下。

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

```

int a,b,t;
scanf("%d%d",&a,&b);
if(a<b)
{
    t=a;a=b;b=t;
}
printf("%d,%d\n",a,b);
}

```

程序运行结果如下。

```

5 12 ✓
12,5

```

4.3.2 if…else 语句

if…else 语句的语法格式如下。

```

if(表达式)
{
    语句块 A
}
else
{
    语句块 B
}

```



视频
else if 语句

该语句用于判断表达式是否成立,若成立则执行语句块 A,不成立则执行语句块 B,如图 4-4 所示。

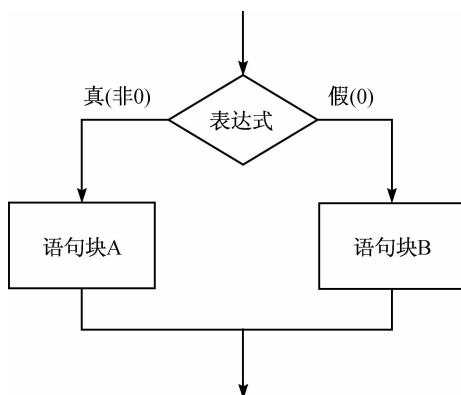


图 4-4 if…else 语句流程图

【例 4-7】 统计及格和不及格的人数。
程序代码如下。

```
if(score>=60)
{
    printf("及格");
    c1=c1+1;
}
else
{
    printf("不及格");
    c2=c2+1;
}
```

上面的程序代码等价于下面的代码。

```
if(score>=60) {printf("及格");c1=c1+1;}
if(score <60) {printf("不及格");c2=c2+1;}
```

【例 4-8】 找 3 个数中的最小数。
程序代码如下。

```
# include <stdio.h>
main()
{
    int a,b,c,min;
    printf("please input a,b,c:");
    scanf(" %d %d %d",&a,&b,&c);
    min=a;
    if(b<min)min=b;
    if(c<min)min=c;
    printf("min= %d\n",min);
}
```

程序运行结果如下。

```
Please input a,b,c:34 ✓
26 ✓
89 ✓
min=26
```

【例 4-9】 3 个数排序。随机输入 3 个数至 a、b、c,编程使得输出 a、b、c 为升序。
程序代码如下。

```
#include <stdio.h>
main()
{
    int a,b,c,t;
    printf("please input a,b,c:");
    scanf("%d%d%d",&a,&b,&c);
    printf("1:a= %d,b= %d,c= %d\n",a,b,c);
    if(a>b) {t=a;a=b;b=t;}
    if(a>c) {t=a;a=c;c=t;}
    if(b>c) {t=b;b=c;c=t;}
    printf("2:a= %d,b= %d,c= %d\n",a,b,c);
}
```

程序的运行结果如下。

```
please input a,b,c:12 8 25 ↵
1:a=12,b=8,c=25
2:a=8,b=12,c=25
```

4.3.3 嵌套 if 语句

一个 if 语句又包含一个或多个 if 语句称为 if 语句的嵌套。嵌套 if 语句的语法格式如下。

```
if(表达式)
    if(表达式)语句 1;
    else 语句 2;
else
    if(表达式)语句 3;
    else 语句 4;
```

这种在 if 语句中又包含 if 语句的选择结构,常用于解决比较复杂的选择问题,其中每一条语句都必须经过多个条件共同决定才能执行。

嵌套 if 语句的几点说明如下。

- (1) 嵌套 if 语句使用非常灵活,任何 if 语句都可以嵌套。
- (2) 被嵌套的 if 语句本身又可以是一个嵌套的 if 语句。
- (3) 在多种嵌套的 if 语句中,else 总是与离它最近并且没有配对的 if 配对。可以用下面两种方法解决匹配问题。

- (1) 利用空语句,使 if 子句数量与 else 子句数量相同。

```
if( )
    if( )语句 1;
```

```

    else;
else
    if( ) 语句 2;
    else 语句 3;

```

(2) 利用 {} 确定配对关系, 将没有 else 子句的 if 语句用 {} 括起来。

```

if( )
{ if( )语句 1;}
else
    if( )语句 2;
    else 语句 3;

```

【例 4-10】 根据下面算式的要求, 编写程序, 输入 x, 输出对应的 y 值。

$$y = \begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$$

程序代码如下。

```

#include <stdio.h>
main()
{
    int x,y;
    scanf(" %d",&x);
    if(x<0)y=-1;
    else if(x==0)
        y=0;
    else y=1;
    printf("x= %d,y= %d\n",x,y);
}

```

程序运行结果如下。

```

8 ✓
x=8,y=1

```

4.3.4 多分支 if…else…if 语句

多分支 if…else…if 语句实际上也是一种特殊的嵌套 if 语句, 不断在 else 子句中嵌套 if 语句可形成多层嵌套。用以下形式表示多分支 if…else…if 语句, 使得程序代码读起来既层次分明, 又不占太多的篇幅。

```

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 n

```

以上嵌套的 if 语句可以理解为:从上到下逐一对 if 后的表达式进行检测。当某一个表达式的值为非 0 时,就执行与此有关子句中的语句,其余部分不执行,直接越过去。如果所有表达式的值都为 0,则执行最后的 else 子句。

【例 4-11】 编写程序,根据输入的学生成绩给出相应的等级,大于或等于 90 分的等级为 A,60 分以下的等级为 E,其余每 10 分为一个等级。

程序代码如下。

```

#include <stdio.h>
main()
{
int score;
printf("Please input score:");
scanf("%d",&score);
if(score<60)
printf("%d-----E\n",score);
else if(score<70)
printf("%d-----D\n",score);
else if(score<80)
printf("%d-----C\n",score);
else if(score<90)
printf("%d-----B\n",score);
else printf("%d-----A\n",score);
}

```

程序运行结果如下。

```

Please input score:87 ✓
87-----B

```

4.4 条件运算符和条件表达式

条件运算符(?)是 C 语言中唯一一个三目运算符。由条件运算符组成条件表达



视频
三目运算符

式的一般形式如下。

```
表达式 1? 表达式 2:表达式 3
```

功能是计算表达式 1, 结果为非零取表达式 2 的值, 为零则取表达式 3 的值。

条件运算符具有左结合性, 优先级低于逻辑“或”(||)运算符, 高于赋值(=)运算符。

例如:

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

等价于:

```
if(a>b) max=a;  
else max=b;
```

【例 4-12】 输入一个字符, 如果是大写字母就转换为小写, 如果不是大写就不转换, 最后输出字符。

程序代码如下。

```
#include <stdio.h>  
main()  
{  
    char ch;  
    scanf("%c",&ch);  
    ch=(ch>='A'&&ch<='Z')?(ch+32);ch;  
    printf("%c",ch);  
}
```

程序运行结果如下。

```
A ✓  
a
```

表达式 `ch=(ch>='A'&&ch<='Z')?(ch+32)` 中的括号都可以不要, 但有括号看上去更加清楚些。

4.5 switch 语句

多分支可以使用嵌套的 if 语句处理, 但如果分支较多, 嵌套的 if 语句层数多, 会使程序冗长, 降低可读性。switch 语句又称为开关语句, 专门用来处理多分支选择问题, 比复合 if 语句及嵌套 if 语句更方便灵活, 而且程序可读性也更高。

switch 语句的语法格式如下。

```

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

```

其含义为:先计算表达式的值,判断此值是否与某个常量的值匹配,如果匹配,控制流程转向其后的语句;否则,检查 default 是否存在,若存在则执行其后的语句,否则结束 switch 语句。switch 语句的流程图如图 4-5 所示。

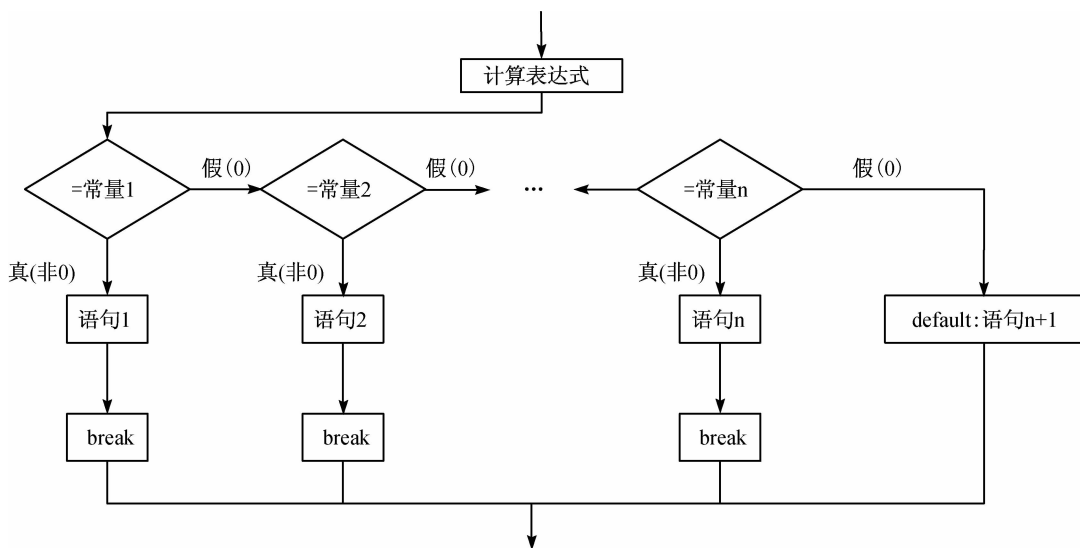


图 4-5 switch 语句流程图

switch 语句的说明如下。

(1) switch 括号后面的表达式允许为任何类型,一般为整型或字符型等有序类型。
 (2) 当“表达式”的值与某个 case 后面的常量表达式的值相等时,就执行此 case 后面的语句。如果表达式的值与所有常量表达式都不匹配,就执行 default 后面的语句(如果没有 default 就执行跳出 switch,执行 switch 语句后面的语句)。

(3) 各个常量表达式的值必须互不相同,否则会出现矛盾。

(4) 各个 case、default 出现的顺序不影响执行结果。

(5) 执行完一个 case 后面的语句后,若子句最后没有 break,流程控制转移到下一个 case 中的语句继续执行。此时,“case 常量表达式”只是起到语句标号的作用,并不在此处进行条件判断。在执行一个分支后,可以使用 break 语句使流程跳出 switch 结构,即终止 switch 语句的执行(最后一个分支可以不用 break 语句)。

注意: switch 语句中本来不包含 break 语句,但 switch 语句不像 if 语句一样只要满足



测试
等级考试真题

某一条件就可在执行相应的分支后自动结束选择。在 switch 语句中,当表达式的值与某个常量表达式的值相等时,就执行后面对应的语句,然后不再进行判断,继续执行后面所有的 case 分支语句,因此需要在相应的 case 分支最后加上 break 以帮助结束选择。

(6) case 后面如果有多条语句,不必用 {} 括起来。

(7) 多个 case 可以共用一组执行语句(注意 break 使用的位置)。

(8) 在关键字 case 和常量表达式之间一定要有空格。

【例 4-13】 使用 switch 语句完成成绩等级的划分。D 不及格(<60)、C 及格($60\sim 79$)、B 良好($80\sim 89$)、A 优秀($90\sim 100$)。

程序代码如下。

```
#include "stdio.h"
main()
{
    float score;char ch;
    scanf("%f",&score);
    switch((int)(score/10.0))
    {
        case 10:
        case 9:  ch='A';break;
        case 8:  ch='B';break;
        case 7:
        case 6:  ch='C';break;
        default: ch='D';
    }
    printf("score= %.1f,grade= %c\n",score,ch);
}
```

程序运行结果如下。

```
78 ↙
score=78.0,grade=C
```

4.6 技能训练

选择结构程序设计是非常重要的程序设计方法,选择结构是大部分程序都涉及的基本结构,在实际应用中被广泛应用。本模块对选择结构控制语句(if 语句、嵌套 if 语句、switch 语句等)进行了介绍。通过本模块的学习,应该了解选择结构程序设计的特点和一般规律,并能够灵活使用 if 语句和 switch 语句设计程序。这就需要多做练习,从每个例题中挑出程序设计的关键语句并熟记,学会正确表达数学问题和实际问题的条件,以便分支处理。

【例 4-14】 编写程序,实现银行根据存入的现金和年限计算利息。

存款年限为 1 年以上时,月息 r 为 5‰。

存款年限为 2 年以上时,月息 r 为 6‰。

存款年限为 3 年以上时,月息 r 为 6.5‰。

存款年限为 5 年以上时,月息 r 为 8‰。

存款年限为 8 年以上时,月息 r 为 10‰。

从键盘输入本金与年限,计算最终资金数。

假定输入的年份为整数,程序代码如下。

```
#include "stdio.h"
main()
{
    int year;
    float money,r,t;
    printf("Please input money and year:");
    scanf("%f%d",&money,&year);
    if(year<0) r=0;
    if(year==1) r=0.005;
    else if(year==2) r=0.006;
    else if(year<5) r=0.0065;
    else if(year<8) r=0.008;
    else r=0.01;
    t=money+money*r*12*year;
    printf("Total=%f\n",t);
}
```

程序运行结果如下。

```
Please input money and year:1234.5 7
Total=2064.08
```

【例 4-15】 某快递公司计算运费的标准如下。路程 s 越远,每千米运费越低。

$s < 250\text{km}$ 没有折扣

$250 \leq s < 500$ 2%折扣

$500 \leq s < 1000$ 5%折扣

$1000 \leq s < 2000$ 8%折扣

$2000 \leq s < 3000$ 10%折扣

$3000 \leq s$ 15%折扣

设每千米每吨货物的基本运费为 p (price),货物重量为 w (weight),距离为 s ,折扣为 d (discount),则总运费 f (freight)的计算公式为:


```
f=p*w*s*(1-d/100.0)
```

分析此问题,折扣的变化是有规律的,250、500、1000、2000、3000 都是 250 的倍数,由此可以得到 $c=s/250$, c 代表 250 的倍数。程序代码如下。

```
#include <stdio.h>
main()
{
    int c,s;
    float p,w,d,f;
    scanf("%f%f%f%d",&p,&w,&s);
    c=s/250;
    switch(c)
    {
        case 0: d=0;break;
        case 1: d=2;break;
        case 2:
        case 3: d=5;break;
        case 4:
        case 5:
        case 6:
        case 7: d=8;break;
        case 8:
        case 9:
        case 10:
        case 11:d=10;break;
        default: d=15;
    }
    f=p*w*s*(1-d/100.0);
    printf("freight= %15.4f",f);
}
```

程序运行结果如下。

```
15 ✓
10 ✓
180 ✓
freight=27000.0000
```

等级考试重难点讲解与真题解析

本模块主要考查选择结构设计方法,包括关系运算符、逻辑运算符、条件运算符以及最常见的 if 条件判断语句。选择结构是 3 种基本的程序设计方法之一。通过对历年试卷内容的分析,考查内容逐年增多。需要重点掌握分支结构的判断条件和执行顺序,选择结构的判断条件一般由关系运算和逻辑运算构成,因此这两个知识点是本模块学习的重点,也是历年考题的重点,一般结合其他知识点综合考核。笔试、上机题均会考核。

一、重点、难点解释

1. 关系运算和逻辑运算

选择结构是按照条件的真假选择执行某段代码。条件中的真(T)和假(F)就是 C 语言中的两种逻辑值,真用非 0 代表,假用 0 代表。关系运算和逻辑运算的结果只有真(非 0)和假(0)两种。

(1)所有运算符的优先级关系是! > 算术运算符 > 关系运算符 > && > || > 赋值运算符,对于有疑问的,请一律用括号来明确。

(2)在 C 程序中,不能出现“连不等式”。形如 $0 < a < 10$ 的不等式一般表示成 $a > 0 \&\& a < 10$ 。

(3)在逻辑表达式的求解中,并不是所有的运算符都被执行,只是在必须执行下一个逻辑运算才能求出表达式解时,才执行下一个运算符。

例如,当 $a=5, b=2$ 时, $a++ || b++$, 因为 $a++$ 已经为真,所以这个时候整个表达式的值已经可以确定为真,不用计算 $b++$ 的值,这个时候 b 的值没有变化。

(4)关于实数相等的比较。在计算机中,通常存放在内存中的实数是有误差的,因此不能精确相等,这将导致 $x=y$ 的值总是假。可以通过 $\text{abs}(x-y) < 0.00001$ 来比较,精度(0.00001)可以由程序员根据要求控制。

2. switch 语句

switch 语句是一种多分支选择语句,其常用形式如下。

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

注意:

(1)各常量表达式的值不能相同,否则会出现错误。

(2)若某子句后没有 break 语句,就按顺序执行下一子句,而不必进行比较判断,直到遇到 break 语句或最后一个“}”时结束 switch 语句。

3. 选择结构的嵌套

if 语句和 switch 语句都可以嵌套使用。在嵌套 if 语句时,要特别注意 else 与哪一个 if 配对。配对原则是:else 总是和最近一个“有资格配对”的 if 配对。在一个 switch 语句中的 case 后面又嵌套了一个 switch 语句,在执行内嵌的 switch 语句后还要执行一条 break 语句才能跳出外层的 switch 语句。

程序段 1:

```
int x=-9,y=-1;
if(x>=0) if(x==0)y=0;
else y=1;
printf(" %d\n",y);
```

程序段 2:

```
int x=-9,y=-1;
if(x>=0) {if(x==0)y=0;}
else y=1;
printf(" %d\n",y);
```

程序段 1 中的 else 与第二个 if 配对,输出的 y 值为 -1;程序段 2 中的 else 与第一个 if 配对,输出的 y 值为 1。

由此可见,用复合语句的方式将不需要配对的 if 语句用 {} 括起来,该 if 语句就被取消了配对资格。

程序段 3:

```
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;
    case 3:a++;b++;
}
printf("a= %d,b= %d\n",a,b);
```

注意程序中第一个 case 后没有 break 语句,程序执行时,x=1,执行内嵌的 switch,因为 y=0,计算 a++,并终止内嵌的 switch,回到外层,继续执行 case 2 后面的语句“a++;

b++”，所以输出结果为 a=2, b=1。

二、真题解析

1. 设有定义“int a=1, b=2, c=3;”，以下语句中执行结果与其他3个不同的是()。
[2009年9月]

- A. if(a>b) c=a, a=b, b=c; B. if(a>b){c=a, a=b, b=c;}
- C. if(a>b) c=a; a=b; b=c; D. if(a>b){c=a; a=b; b=c;}

解析 “if(a>b) c=a; a=b; b=c;”先执行“if(a>b) c=a;”，因为 a<b，所以不执行 c=a，然后顺序执行“a=b; b=c;”，所以 C 的执行结果为 a=2, b=3, c=3。其他3个选项 a、b、c 的值均为 2、1、1。

答案 C

2. if 语句的基本形式是:if(表达式)语句, 以下关于“表达式”值的叙述中正确的是()。[2011年3月]

- A. 必须是逻辑值 B. 必须是整数值
- C. 必须是正数 D. 可以是任意合法的数值

解析 if(表达式)语句中的表达式是逻辑表达式, 逻辑运算符两侧的对象不但可以是 0 和 1 或者是 0 和非 0 的整数, 也可以是任何类型的数据(如字符型、实型、指针型)。

答案 D

3. 若变量已正确定义, 在“if(w)printf("%d\n, k);”中, 以下不可替代 w 的是()。
[2010年9月]

- A. a<>b+c B. ch=getchar()
- C. a==b+c D. a++

解析 在 C 语言中没有<>这种形式的运算符。

答案 A

4. 若有表达式(w)?(--x):(++y), 则下列与 w 等价的表达式是()。[2008年9月]

- A. w==1 B. w==0 C. w!=1 D. w!=0

解析 本题考查了逻辑表达式的真假。题目中若 w 为非 0, 表达式 w 表示真, w 为 0, 表达式 w 表示假。

答案 D

5. 执行以下程序段后, w 的值为()。[2008年9月]

```
int w='A', x=14, y=15;
w=((x||y)&&(w<'a'));
```

- A. -1 B. NULL C. 1 D. 0

解析 本题考查了逻辑表达式的判断。只有 && 两边都为真, 表达式才为真; || 两边有一个为真就为真; (x||y) 值为真, (w<'a') 也为真, 整个表达式((x||y)&&(w<'a')) 值为 1, 所以 w 的值为 1。

答案 C

6. 有以下程序。

```
#include<stdio.h>
main()
{
    int a=1,b=0;
    if(! a) b++;
    else if(a==0)if(a)b+=2;
        else b+=3;
    printf("%d\n",b);
}
```

程序运行后的输出结果是()。[2010年9月]

- A. 0 B. 1 C. 2 D. 3

解析 本题考查 if...else 的嵌套。! a 为假,执行 else 后面的 if 语句,a==0 为假,后面的 if...else 不再执行。整个过程中 b 均未再赋值,故仍保持原值为 0。注意最后一个 else 是与 if(a) 配对的。

答案 A

7. 若有定义语句“int a, b;double x;”,则下列选项中没有错误的是()。[2010年9月]

- | | |
|---|---|
| <p>A. switch(x%2)</p> <pre>{ case 0: a++; break; case 1: b++; break; default : a++; b++; }</pre> | <p>B. switch((int)x/2.0)</p> <pre>{ case 0: a++; break; case 1: b++; break; default : a++; b++; }</pre> |
| <p>C. switch((int)x%2)</p> <pre>{ case 0: a++; break; case 1: b++; break; default : a++; b++; }</pre> | <p>D. switch((int)(x)%2)</p> <pre>{ case 0.0: a++; break; case 1.0: b++; break; default : a++; b++; }</pre> |

解析 本题考查 switch 语句。switch 后面括号内表达式的值必须为有序类型,case 后面必须是对应类型的常量表达式。符合要求的只有 C 选项。

答案 C



思考与练习

一、选择题

1. 下列选项中,不能看做一条语句的是()。

A. {;} B. if(b==0)m=1;n=2;

C. if(a>0); D. a=0,b=0,c=0;

2. 下列运算符中优先级最低的是()。

A. || B. != C. <= D. +

3. 设 a、b 和 c 都是 int 型变量,且 a=3,b=4,c=5,则以下表达式中,值为 0 的表达式是()。

A. a&&.b B. a<=b C. a||b+c&&.b-c D. !((a<b)&&! c||1)

4. 有以下程序。

```
main()
{
    int a=2,b=-1,c=2;
    if(a<b)
        if(b<0) c=0;
    else c+=1;
    printf("%d\n",c);
}
```

程序的输出结果是()。

A. 1 B. 0 C. 2 D. 3

5. 以下程序执行的结果是()。

```
main()
{
    int w=4,x=3,y=2,z=1;
    printf("%d\n",(w<x? w:z<y? z:x));
}
```

A. 1 B. 2 C. 3 D. 4

6. 有以下程序。

```
main()
{
    int a, b,s;
    scanf("%d%d",&a,&b);
    s=a;
    if(a<b) s=b;
    s*=s;
    printf("%d\n",s);
}
```

若执行程序时从键盘输入 3 和 4,则输出结果是()。

- A. 14 B. 16 C. 18 D. 20

二、填空题

1. C 语言中用 _____ 表示逻辑值真,用 _____ 表示逻辑值假。
2. 已知 $a=2.3, b=2, c=3.6$, 表达式 $a>b \&\&c>a || a<b$ 的值是 _____。
3. 当 $a=11, b=22, c=32$ 时, 以下 if 语句执行后, a, b, c 的值分别为 _____、_____、_____。

```
if(a>c)
```

```
    b=a;a=c; c=b;
```

4. 以下程序的输出结果是 _____。

```
main()
{
    int x=1,y=0;
    if(x=y)
        printf("MYMMYMMY");
    else
        printf(" * * * ");
}
```

三、编程题

1. 编写程序,输入一位学生的生日(年:y0、月:m0、日:d0);并输入当前的日期(年:y1、月:m1、日:d1);计算并输出该学生的实际年龄。
2. 编写程序,输入一个整数,打印出它是奇数还是偶数。
3. 设计简易计算器。根据输入的两个运算数和一个运算符进行相关的运算。
4. 编程实现根据某人的身高和体重判断其身体指数。
身体指数与体重、身高的关系为:
身体指数 $t=w/h^2$ (w 表示体重,单位为 kg,h 表示身高,单位为 m)
当 $t<18$ 时,偏瘦。
当 $18 \leq t < 25$ 时,正常体重。
当 $25 \leq t < 27$ 时,超重。
当 $t \geq 27$ 时,肥胖。
5. 编写程序,输入某年某月,求该月的天数。