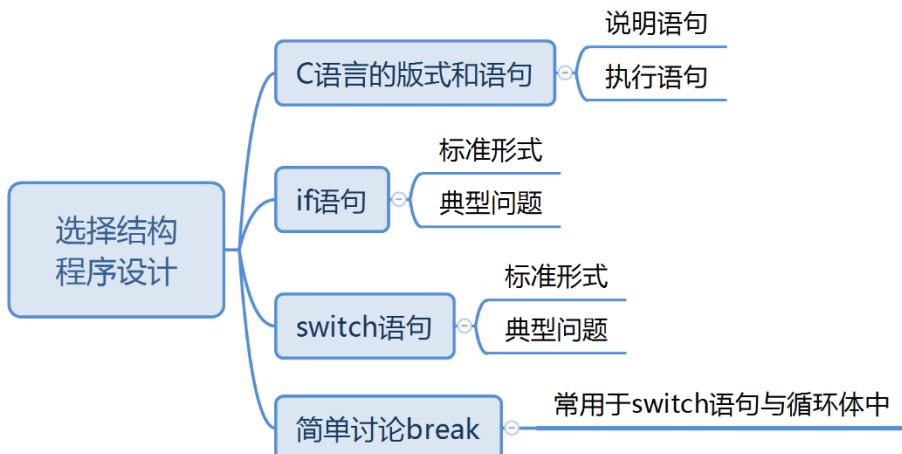




专题三 选择结构

【内容预览】



【知识清单】

3.1、C语言的版式和语句

3.1.1、C语言程序的版式

- 举一个简单的例子进行分析：

要求：写一段求两整数之积的程序。

```

#include<stdio.h>           //预处理
int multiply(int x,int y);   //函数声明（关于函数的部分，在后面的章节会细讲，在此就不赘述）
void main()                  //函数定义
{
    int x,y;
    int mul;                 //变量说明
    scanf("%d%d",&x,&y);     //数据的输入
    mul=multiply(x,y);       //执行部分
    printf("mul=%d",mul);    //信息输出
}
int multiply(int x,int y)
{
    int z;
  
```

```

z=x*y;
return z;
}

```

2. 编写的 C 语言程序一般应包含如下几个部分。

- (1) **注释部分**: 格式为“/*注释内容*/”或者“//注释内容”。在函数的最上端，一般都应有一段注释信息，主要说明函数的功能，输入、输出及其限制。
- (2) **预处理模块、全局变量说明等**(这个在后面的章节中会详细介绍)
- (3) **函数定义部分**: 包括函数返回值的类型，函数名及参数表。由于只有一个主函数故取名 main(); 由于无返回值故类型为 void。在无参数输入时，main()内参数表为空。
- (4) **变量说明部分**: 对所用的变量进行说明。
- (5) **数据输入部分**: 对变量赋初值，可直接或间接输入，有些是在第（4）部分完成（对变量直接初始化。例如：int i=0;）
- (6) **执行部分**: 一般是对程序算法用程序化的设计方法进行描述。
- (7) **信息输出部分**: 根据所需要的数据进行结果输出。

3.1.2、C 语言基本语句

1. **空语句**: 只有 1 个分号的语句，写成： ; ，1 条语句的最后一般用分号结束，分号是一条语句结束的标志。
2. **表达式语句**: 任意一个表达式后加 1 个分号构成的 1 条语句。如： m=0 是 1 个赋值表达式，表达式将 0 赋给变量 m，且整个表达式有值（为 0）。而 m=0; 是 1 个赋值语句，表示执行 1 个赋值过程。
3. **函数调用语句**: 由 1 个函数调用语句加 1 个分号构成。
如： printf("this is an example.\n");
4. **控制语句**

条件控制语句	循环控制语句	辅助控制语句
if 语句 switch 语句	while 语句 do-while 语句 for 语句	break 语句 continue 语句 goto 语句 return 语句

表 3-1 控制语句分类

内嵌语句: 条件控制语句和循环控制语句中均可再含有语句，一般把包含在条件控制语句和循环控制语句中的语句称为内嵌语句。

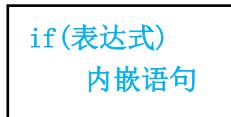
5. **复合语句**: 若将顺序执行的语句序列用大括号括起来，就构成 C 语言的复合语句，在逻辑上视复合语句为 1 个单一语句块，它也能用作其它结构的成分语句。程序中通常用复合语句描述顺序结构。

3.2、if 语句

3.2.1、if 语句标准形式

1. 单分支的 if 语句

(1) 格式:

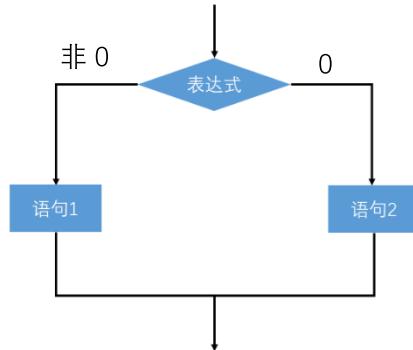
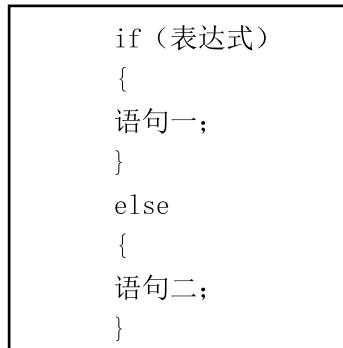


(2) 说明:

- a) 若内嵌语句中没有强制性的语句（如：`goto`、`continue`、`break` 等），则执行内嵌语句后再执行 if 的下一条语句。
- b) 在 C 语言中，if 语句中的条件是用表达式来反应的，并且只关心表达式的值是 0 还是非 0. 非 0 就认为是条件满足，执行内嵌语句。
- c) 注意以下几种情况的区别
 - ◆ 若有 `if(a>b);a++;b++;`，其内嵌语句是空语句 `;`，`a++;b++;` 是 if 的后续语句。语句 `if(a>b);` 不论是否为真，都会执行 `a++;` 和 `b++;` 语句。
 - ◆ 若有 `if(a>b){a++;b++};`，则 `a++;b++;` 都是其内嵌语句。当 `a>b` 时，语句都会执行。
 - ◆ 若有 `if(a>b)a++;b++;`，则 `a++;` 是其内嵌语句，而 `b++;` 是 if 的后续语句。

2. 双分支的 if 语句

(1) 其一般形式为：



(2) if 后面圆括号中的表达式一般是关系表达式，它表示分支的条件，这些会在后面的学习中遇到，这里举一个常见的例子。

```

if(a>=0)
{
    printf("Hey,boy");
}
else
{
    printf("Hey,guy");
}
    
```

偶尔题目中也会出现一些赋值语句，如：if(a=5)，这个很容易让做题者认为是对 a 是否等于 5 进行判断，然而实际上，这个是一个赋值语句，无论如何，if 中的条件都成立，都可以执行 if 后面的内嵌语句。

```
如： void main()
{
    int a=0;
    if(a=5)
        printf("%d",a);
    else printf("%d",a*a);
}
```

输出结果为：5

3. 嵌套的 if 语句

一般形式如下：

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

例如：if (number>500) cost=0.15;
else if(number>300) cost=0.10;
else if(number>100) cost=0.075;
else if(number>50) cost=0.05;

说明：(1) 在 if 嵌套结构，每个 else 总要与它前面的 if 对应，**不能够出现没有对应 if 的 else,else 总是与它前面最接近的且没有匹配的 if(表达式)相配对。**

(2) 若不希望与最近的 if 语句配对，可用大括号{}把该 if 语句括起来。

3.2.2 if 语句典型例题

1. if 条件语句可以用来表示符号函数 $y=\begin{cases} 1 & (x>0) \\ 0 & (x=0) \\ -1 & (x<0) \end{cases}$

程序可以写成：

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

```

int x,y;
scanf("%d",&x);//注意输入函数的格式
if(x<0)      //判断 x, 来决定 y 的值是取 1、0 还是-1。
y=-1;
else if(x==0)
y=0;
else
y=1;    }

```

2. 给某班的学生的一次考试成绩分等级，其中，i 表示学生成绩，grade 表示等级；90 分以上为 A，70 分到 90 分之间为 B，60 到 70 之间为 C，60 分以下的得 D。

```

#include<stdio.h>
void main()
{
    int i;
    char grade;
    scanf("%d",&i);
    if(i>=90)
        grade='A';
    else if(i>=70)
        grade='B';
    else if(i>=60)
        grade='C';
    else
        grade='D';
}

```

【解题技巧】

例 3.2.1 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（ ）

- | | |
|-----------|---------------|
| 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 |

正解：D

例 3.2.2 设有定义：int a=1,b=2,c=3;以下语句中执行效果与其它 3 个不同的是（ ）

- | | |
|------------------------|--------------------------|
| 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;} |

正解: C

分析: A、B、D 都表示若 $a > b$; 则执行其后的 $c = a$ 、 $a = b$ 和 $b = c$ 语句, 而 C 中表示, 若 $a > b$, 则执行 $c = a$; 语句, 而 $a = b$; $b = c$ 语句的执行不受 if 条件语句的限制。

3.3、switch 语句

3.3.1、switch 语句标准形式

- if 语句只有两个分支可供选择, 虽然可以用嵌套的 if 语句来实现多分支选择, 但是**当在处理分支较多而且复杂的情况时, switch 语句的优越性就能体现出来**。适当地使用 switch 语句可能会让程序大大优化, 使结构更加清晰。其一般形式为:

```
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;
}
```

- switch 后面的表达式可以是数值型也可以是字符型表达式; case 后面的常量表达式只能是整型常量、字符常量或符号常量组成的表达式。

说明:

- case 后面的常量表达式中不能有变量或函数;
- case 执行语句末尾若没有 “break;”, 则执行完该部分后继续执行之后的 case 的执行语句;
- 若 switch 中表达式的值不是整数则自动取整;
- 每个常量表达式的值必须互不相同;
- case 和 default 后的语句可以是单条语句, 也可以是复合语句 (仅在这里的复合语句可以不用大括号括起来);
- 常量表达式后的语句可缺省, 使多个 case 共用一组执行语句;
- 某个常量表达式一旦匹配将执行其后的所有语句, 不再判别是否与其它常量表达式相匹配;
- 若缺省 default 项, 当所有常量表达式都不匹配时, 则不执行任一条语句, 直接执行该结构的下一条语句;
- 若结构含有 default 项, 该项可以置于 case 之间 (即 default 项可以不是结构的最后一项)。同 f)一样, default 后 case 中的语句都将被执行;
- switch 语句的判断选择是没有优先级的, if 条件语句判断是有优先级的。

3.3.2、switch 语句典型例题

1. 用 switch 语句可以用来表示函数 $y = \begin{cases} x+1 & 0 \leq x < 2 \\ x+2 & 2 \leq x < 4 \\ x+3 & 4 \leq x < 6 \\ x+4 & 6 \leq x < 8 \end{cases}$

```
#include<stdio.h>
void main()
{
    float x,y;
    scanf("%f",&x);
    switch((int)x) //由于前面定义了 float 型的变量 x, 所以这里需要强制转换为 int 型,
                    //但是 x 的值并不会发生变化。
    {
        case 0:
        case 1:
            y=x+1; //如果两条 case 语句后面执行的指令相同, 可以采用如上形式,
                    //将两条语句合并。
            break;
        case 2:
        case 3:
            y=x+2;
            break;
        case 4:
        case 5:
            y=x+3;
            break;
        case 6:
        case 7:
            y=x+4;
            break;
        default:break; //这里的 defult 后面执行的是空语句, 可视为不进行任何操作。
    }
}
```

2. 给某班的学生的一次考试成绩分等级, 其中, i 表示学生成绩, grade 表示等级; 90 分以上为 A, 70 分到 90 分之间为 B, 60 到 70 之间为 C, 60 分以下的得 D。

这道题在上一小节中已经用 if-else 语句进行了编写, 我们可以来回顾一下:

```
#include<stdio.h>
void main()
{
    int i;
    char grade;
    scanf("%d",&i);
    if(i>=90)
        grade='A';
    else if(i>=70)
        grade='B';
    else if(i>=60)
```

```

        grade='C';
    else
        grade='D';
    printf("%c",grade);
}

```

可以看出，用 if-else 语句编写的程序用了很多 if 语句的嵌套，错综复杂，下面来看一下用 switch 编写的程序。

```

#include<stdio.h>
void main()
{
    int i;
    char grade;
    scanf("%d",&i);
    switch(i/10)           //由于 i 本就是 int 型，所以这里不用强制转换。
    {
        case 9:
            grade='A';
            break;
        case 8:
        case 7:
            grade='B';
            break;
        case 6:
            grade='C';
            break;
        default:
            grade='D';
    }
    printf("%c",grade);
}

```

由上面的代码可以看出，虽然比用 if-else 语句编写程序要长一些，但是条理更加清晰。

3. 用 switch 编写一个简单的四则运算程序

```

#include<stdio.h>
int main()
{double a,b,c;
char op;
scanf("%lf%c%lf",&a,&op,&b);
switch(op)
{case '+': c=a+b; break;
case '-': c=a-b; break;
case '*': c=a*b; break;
case '/': if(b==0) printf("除数不能为 0!\n"); return 0;
else {printf("除数不能为 0!\n"); return 0;}
default:{printf("未知的运算符!\n"); return 0;}
}

```

```

printf("%g%c%g=%g\n",a,op,b,c);
return 0;
}

```

【解题技巧】

例 3.3.1 有以下程序：

```

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

```

程序的运行结果是（ ）

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

正解：B

分析：在第一个 switch 语句处进行判断，执行 case 1:；再进行第二个 switch 语句，进行 case 0:语句得 a=1；第一个 switch 的 case 1: 后无 break; 语句，所以，继续执行第一个 switch 语句下的 case 2: 语句，得到 a=2, b=1；执行 break 语句，结束。

3.4、对 break 的简单讨论

3.4.1、break 的用法讨论

1. break 语句通常用在**循环语句**和**switch 语句**中。当 break 语句用于开关语句 switch 中时，可使程序跳出 switch 而执行 switch 以后的语句；**如果没有 break 语句，则会从当前的 case 语句组一直向下运行，直到把每一条都运行结束**，往往这并不是我们所需要的结果。
2. **当 break 语句用于 do-while、for、while 循环语句中时，可使程序终止该层循环而执行外层循环或循环后面的语句**，通常 break 语句总是与 if 语句联在一起。即满足条件时便跳出循环。（此条会在第四章再进行讲解）

【精选习题】

答案 P112

基础篇

1. 当变量 c 的值不为 2、4、6 时，值为“真”的表达式是（ ）

- A) $(c >= 2 \&\& c <= 6) \&\& (c \% 2 != 1)$
- B) $(c == 2) || (c == 4) || (c == 6)$
- C) $(c >= 2 \&\& c <= 6) \&\& !(c \% 2)$
- D) $(c >= 2 \&\& c <= 6) || (c != 3) || (c != 5)$

2. 有嵌套的 if 语句：

```
if(a<b)
    if(a<c)k=a;
    else k=c;
else if(b<c)k=b;
else k=c;
```

以下选项中与上述 if 语句等价的语句是（ ）

- A) $k = (a < b) ? ((b < c) ? a : b) : ((b > c) ? b : c);$
- B) $k = (a < b) ? ((a < c) ? a : c) : ((b < c) ? b : c);$
- C) $k = (a < b) ? a : b; k = (b < c) ? b : c;$
- D) $k = (a < b) ? a : b; k = (a < c) ? a : c;$

3. 下列叙述中正确的是（ ）

- A) 在 switch 语句中不一定使用 break 语句
- B) 在 switch 语句中必须使用 default 语句
- C) break 语句必须与 switch 语句中的 case 配对使用
- D) break 语句只能用于 switch 语句

4. 设有定义： int a=2,b=3,c=4;; 则以下选项中值为 0 的表达式是（ ）

- A) $a \&\& b$
- B) $a < b \&\& !c || 1$
- C) $(!a == 1) \&\& (!b == 0)$
- D) $a || (b + b) \&\& (c - a)$

5. #include<stdio.h>

```
void main()
{
    int x;
    scanf("%d", &x);
    if(x++ > 5)
        printf("%d\n", x);
    else
        printf("%d\n", x--);
}
```

若输入整数为 5，程序运行的结果是： _____

6. 写出下面程序执行后的结果

```
#include <stdio.h>
void main()
{
    int k=5,n=0;
    while(k>0)
    {
```

```

switch(k)
{
    default : break;
    case 1 : n+=k;
    case 2 :
    case 3 : n+=k;
}
k--;
}

printf("%d\n",n);
}

```

结果为： _____

提高篇

1. 有以下程序：

```

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

```

程序的运行结果是（ ）

- A) 1,3,2 B) 1,3,3 C) 1,2,3 D) 3,2,1

2. 有以下程序：

```

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

```

程序运行的结果是（ ）

- A) 1,1,2,0 B) 0,0,0,3 C) 编译有错 D) 0,1,2,0

3. 若有定义；float x=1.5;int a=1,b=3,c=2;;则正确的 switch 语句是（ ）

- | | |
|---|---|
| <p>A) switch(a+b)</p> <pre>{ case 1:printf("*\n"); case 2+1:printf("**\n"); }</pre> | <p>B) switch(int(x));</p> <pre>{ case 1:printf("*\n"); case 2:printf("**\n"); }</pre> |
| <p>C) switch(x)</p> <pre>{ case 1.0:printf("*\n"); }</pre> | <p>D) switch(a+b)</p> <pre>{ case 1:printf("*\n"); }</pre> |

case 2.0:printf("**\n"); } case c:printf("**\n"); }

4. 下列选项中与 if(a==1)a=b;else a++;语句功能不同的 switch 语句是 ()

- | | |
|----------------------|---------------------|
| A) switch(a==1) | B) switch(a) |
| { case 0:a=b;break; | { case 1:a=b;break; |
| case 1:a++; } | default :a++; } |
| C) switch(a) | D) switch(a==1) |
| { default:a++;break; | { case 1:a=b;break; |
| case 1:a=b; } | case 0:a++; } |

5. 读程序，写出结果。

```
main()
{ int a=0,b=1,c=0,d=20;
  if(a) d=d-10;
  else if(!b)
    if(!c)d=15;
    else d=25;
  printf("d=%d\n",d);
}
```

结果为: _____

6. #include<stdio.h>

```
void main()
{ int a=5,b=4,c=3,d=2;
  if(a>b>c) printf("%d\n",d);
  else if((c-1>=d)==1) printf("%d\n",d+1);
  else printf("%d\n",d+2);
}
```

程序运行的结果是: _____