



计算机导论与程序设计——第7篇

基本程序控制结构

Computer Introduction and Programming

学习目标



- 熟悉结构化程序设计概念
- 掌握关系表达式和条件表达式
- 掌握分支结构及相关语法：if语句和switch语句；
- 掌握循环结构及相关语法：for语句；while语句；do-while语句；嵌套循环；break和continue用法；
- 掌握多种控制结构嵌套使用方法

C程序结构

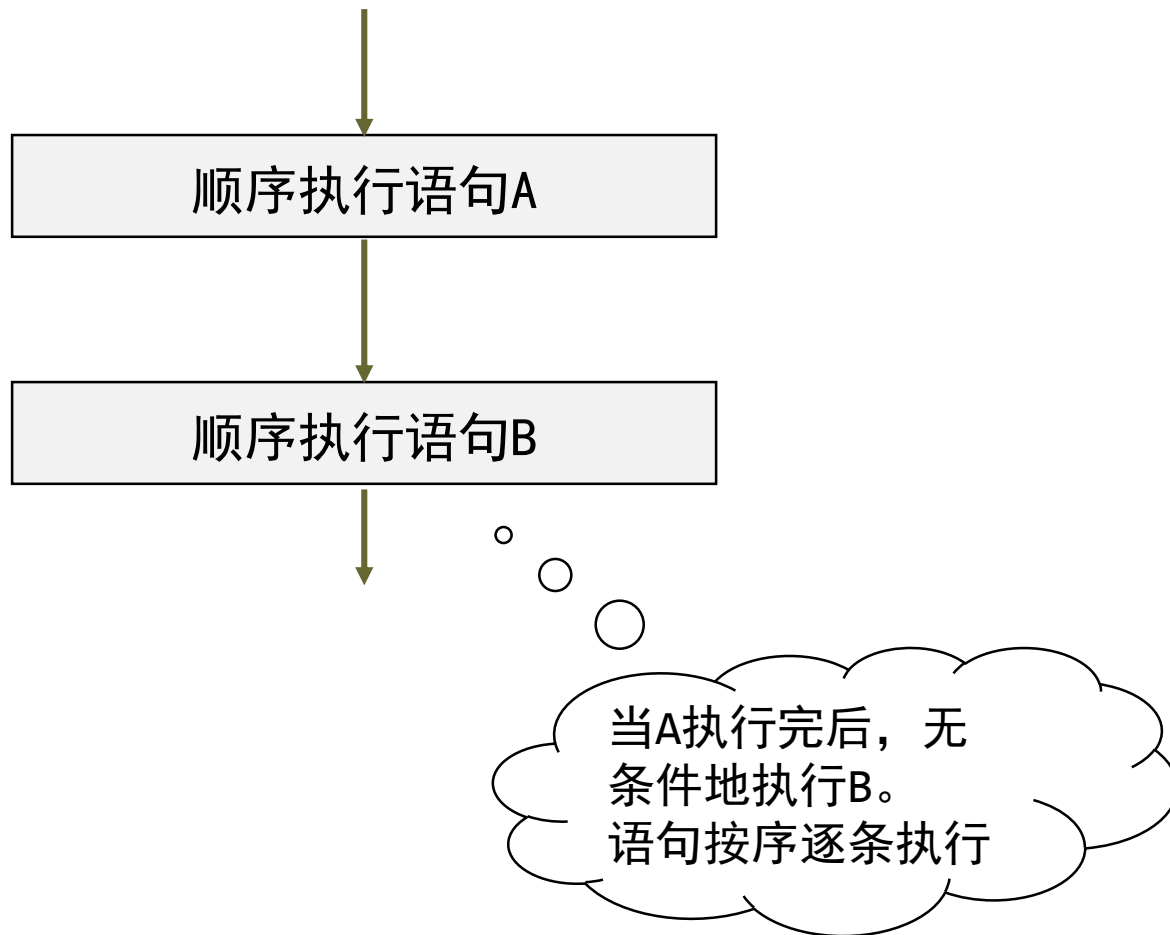


顺序结构

选择结构

循环结构

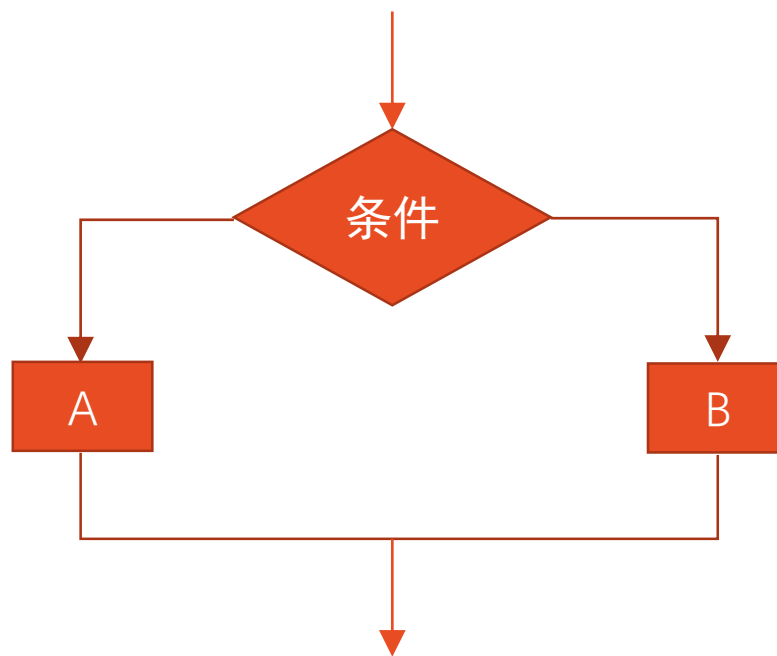
顺序结构的流程图表示



选择结构

选择结构

对于选择程序结构，先判断给定的条件，再根据判断的结果来控制程序的流程。



if语句的一般形式

if (表达式) 语句1
[else 语句2]

“表达式”的值为“真”或者“假”

方括号内的部分(即else子句)为可选的，既可以有，也可以没有

语句1和语句2可以是一个简单的语句，也可以是一个复合语句

形式1 没有else子句部分

if(表达式) 语句1

形式2 有else子句部分

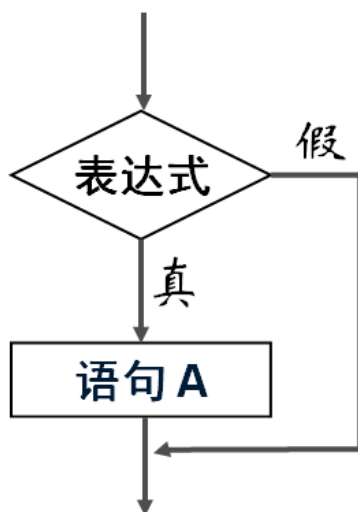
if (表达式)
 语句1
else
 语句2

形式3 在else部分又嵌套了多层的if语句

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

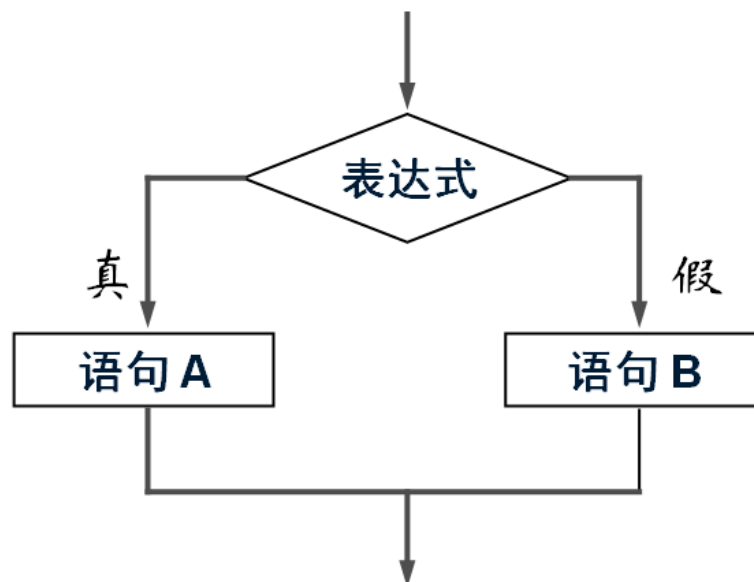
选择结构的流程图表示法

单路选择



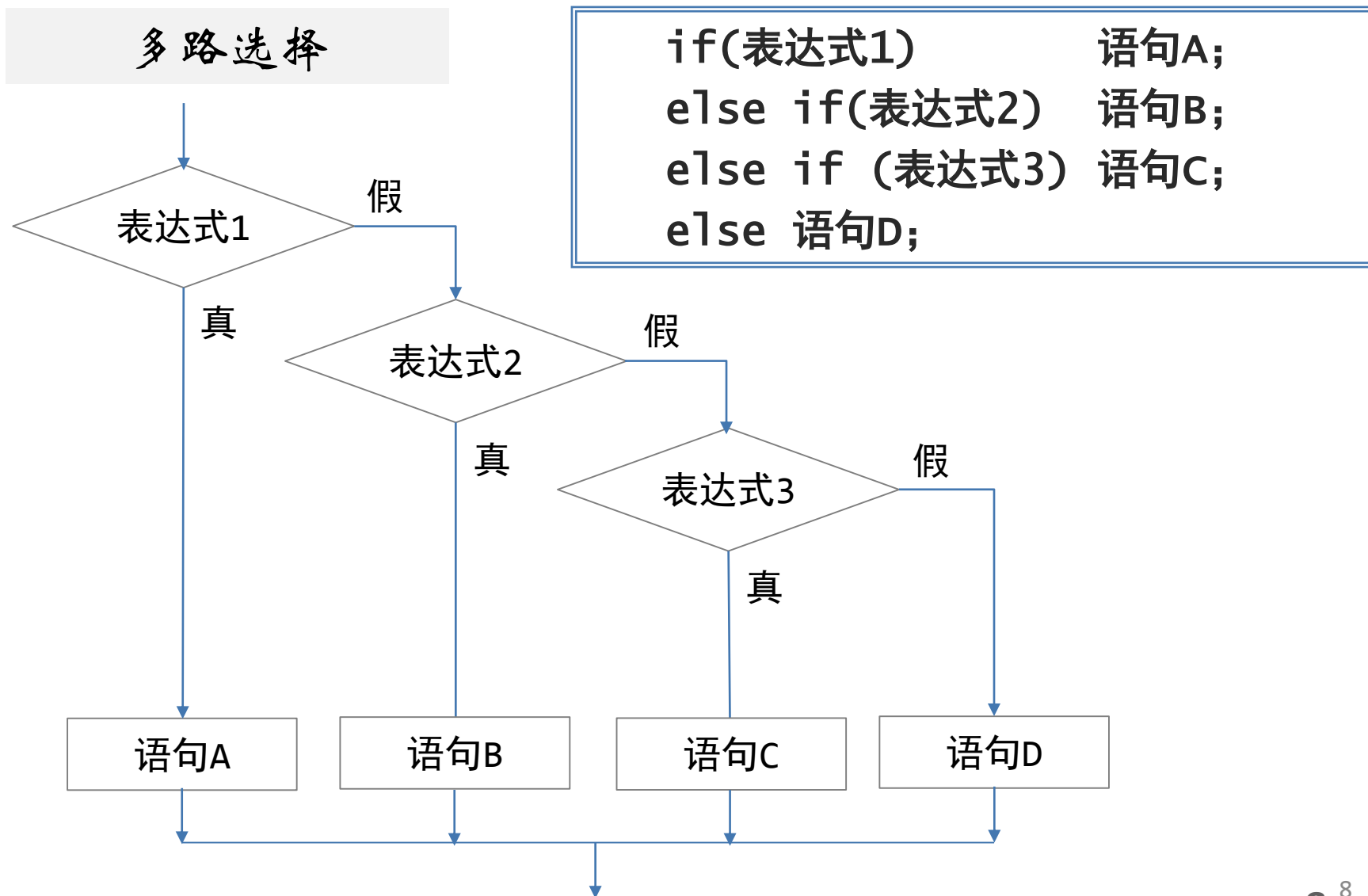
if (表达式) 语句 A;

双路选择



if (表达式) 语句 A;
else 语句 B;

选择结构的流程图表示法



C的条件语句

种类	语句		含义
单路选择	if语句	if (表达式) 语句 A;	表达式成立, 执行语句A
双路选择		if (表达式) 语句 A; else 语句 B;	表达式成立, 执行语句A 否则, 执行语句B
多路选择		if (表达式1) 语句 A; else if(表达式2) 语句 B; else 语句 C;	表达式1成立, 执行语句A; 否则, 若表达式2成立, 执行 语句B; 否则, 执行语句C

if语句的嵌套

如果在判断结构中需要进行多分支选择，那么可以使用嵌套的if~else结构，即在if~else结构中的语句1或语句2中，又包含一个if~else结构。

```
if ( ) {  
    if ( ) 语句;  
    else 语句;  
} else 语句;
```

复合语句

```
if ( ) 语句;  
else {  
    if ( ) 语句;  
    else 语句;  
}
```

复合语句

用 { } 括号括起来
的一组语句



规则

if语句的嵌套结构中，从内层开始，else总是与上面最近的(未曾配对的)if匹配。

if语句的表达式

形式1 没有else子句部分

```
if(表达式) 语句1
```

形式2 有else子句部分

```
if (表达式)
    语句1
else
    语句2
```

形式3 在else部分又嵌套了多层的if语句

```
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
```

“表达式”可以是：

- 关系表达式
- 逻辑表达式
- 数值表达式

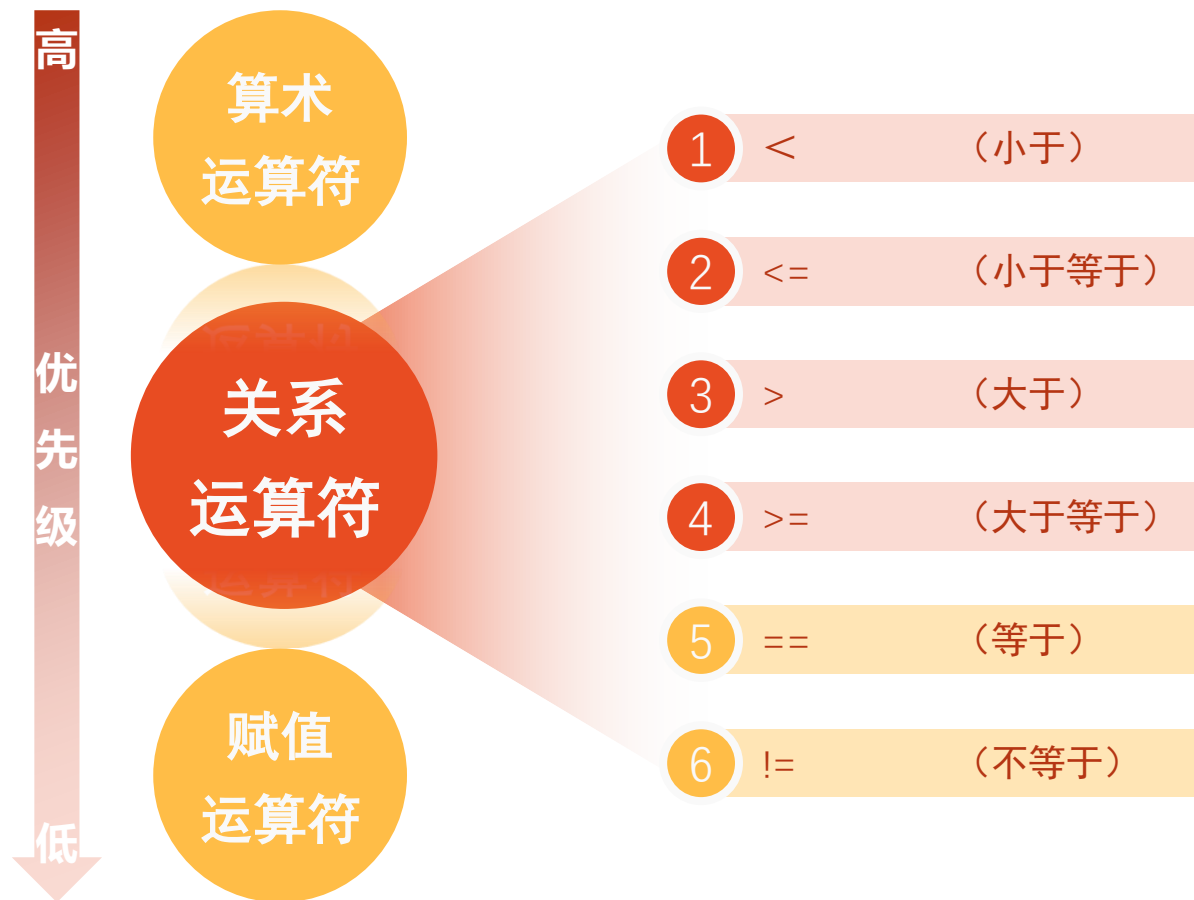


关系运算符和关系表达式

- 比较运算符称为关系运算符。
- “关系运算”就是“比较运算”，将两个数值进行比较，判断其比较的结果是否符合给定的条件。



关系运算符及其优先次序



- 关系表达式的值是一个逻辑值，即“真”或“假”。
- 在C的逻辑运算中，以“1”(或非0)代表“真”，以“0”代表“假”。

逻辑运算符和逻辑表达式

运算符	含义	举例	说明
&&	逻辑与 (AND)	a && b	如果a和b都为真，则结果为真，否则为假
	逻辑或 (OR)	a b	如果a和b有一个以上为真，则结果为真，二者都为假时，结果为假
!	逻辑非 (NOT)	!a	如果a为假，则!a为真;如果a为真，则!a为假

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。

逻辑表达式的练习

判断用整型变量year表示的某一年是否是闰年。

闰年的条件是符合下面二者之一：1) 能被4整除，但不能被100整除，如2008；2) 能被400整除，如2000。写出表达式

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

if语句的例子1

	取值	条件
y=	1	x > 2
	2	其余

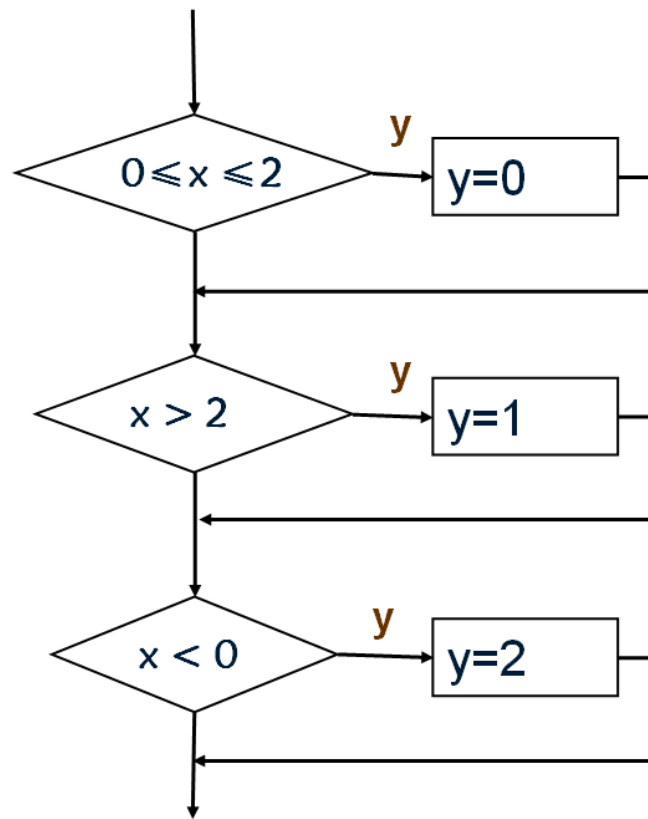
```
if(x > 2)
    y = 1;
else
    y = 2;
```


if语句的例子2

	取值	条件
y=	0	$0 \leq x \leq 2$
	1	$x > 2$
	2	$x < 0$

方法一

```
if(x >= 0 && x <= 2) y = 0;  
if(x > 2) y = 1;  
if(x < 0) y = 2;
```

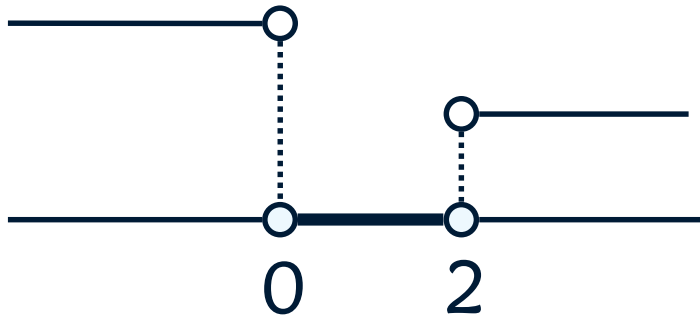


评价：可读性好,效率不高。

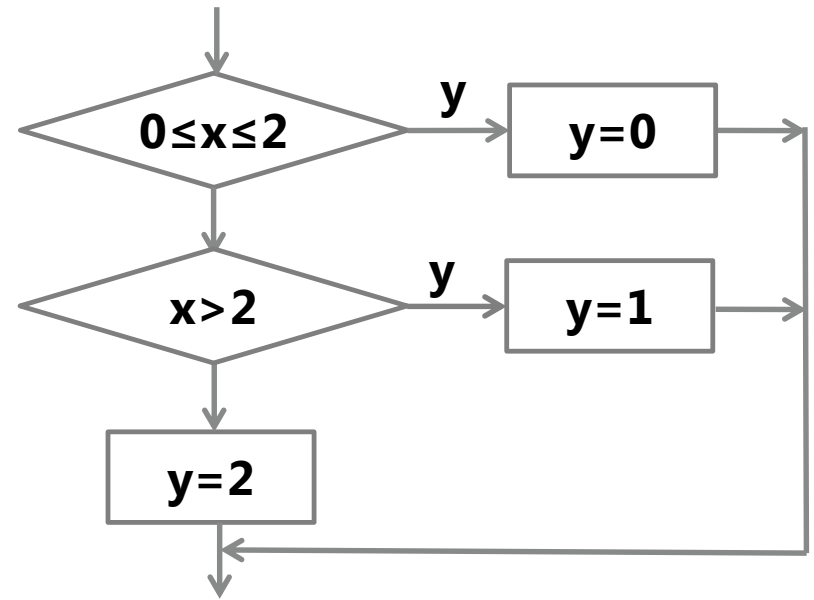
if语句的例子2

方法二

	取值	条件
y=	0	$0 \leq x \leq 2$
	1	$x > 2$
	2	$x < 0$



评价：效率提高，可读性不好



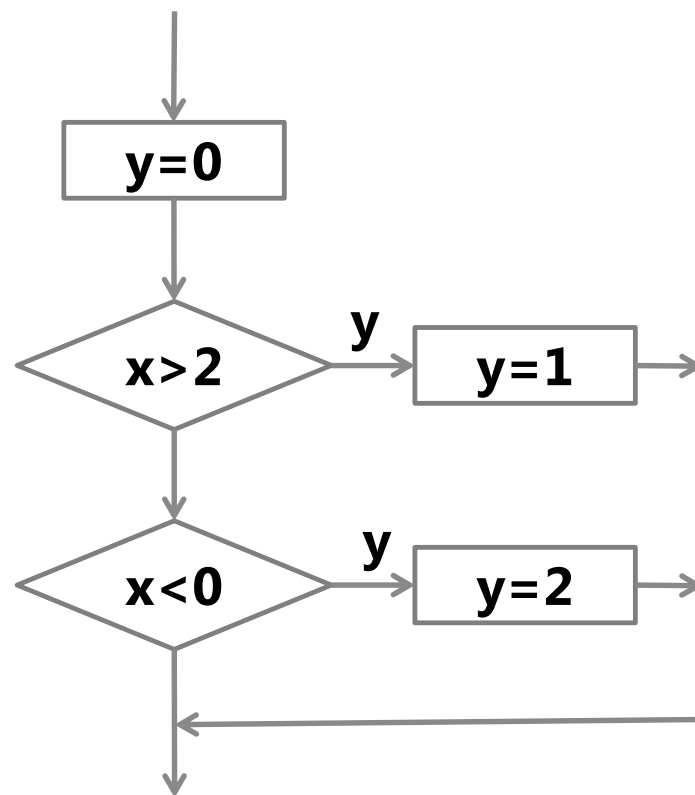
```
if(x>=0 && x<=2)
    y=0;
else{
    if(x>2)
        y=1;
    else
        y=2;
}
```

if语句的例子2

方法三

	取值	条件
y=	0	$0 \leq x \leq 2$
	1	$x > 2$
	2	$x < 0$

```
y=0;  
if(x>2)  
    y=1;  
else if(x<0)  
    y=2;
```



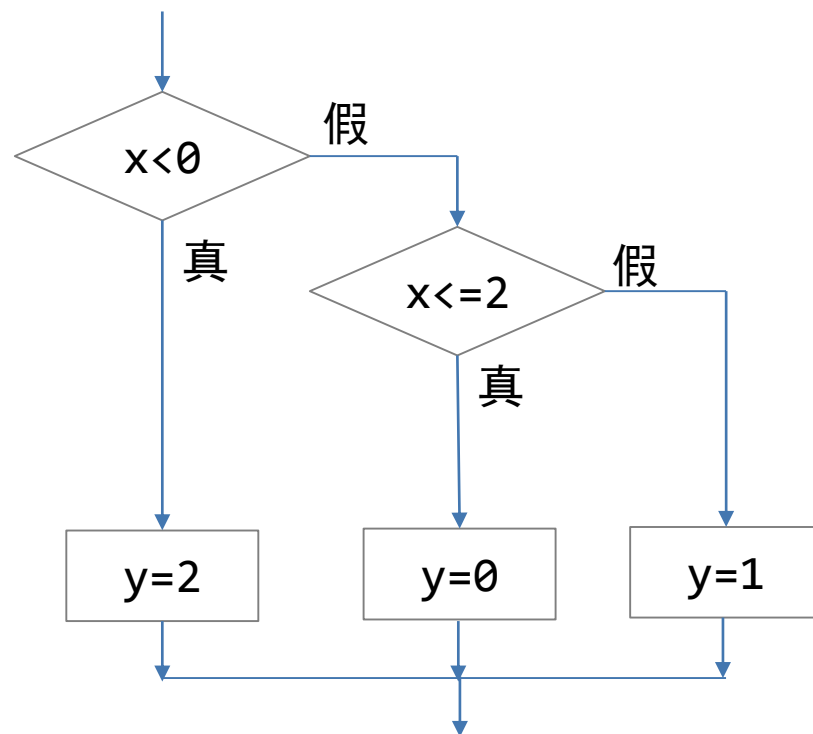
评价：效率高，可读性较好

if语句的例子2

方法四

	取值	条件
y=	0	$0 \leq x \leq 2$
	1	$x > 2$
	2	$x < 0$

```
if(x < 0)
    y = 2;
else if (x <= 2)
    y = 0;
else
    y = 1;
```



评价：效率高，可读性好

【例7.1】求解 $ax^2+bx+c=0$ 方程的根。如果 $b^2-4ac \geq 0$ ，就计算并输出方程的两个实根，如果 $b^2-4ac < 0$ ，就输出“此方程无实根”的信息。

```
1 #include<stdio.h>
2 #include<math.h>//程序中要调用求平方根函数sqrt
3 int main() {
4     double a,b,c,disc,x1,x2,p,q;//disc是判别式sqrt(b*b-4ac)
5     scanf("%lf%lf%lf",&a,&b,&c);//输入双精度要用格式声明"%lf"
6     disc=b*b-4*a*c;
7     if (disc<0) //若b*b-4ac<0, 输出“此方程无实根”
8         printf("This equation hasn't real roots\n");
9     else { //b*b-4ac≥0
10         p=-b/(2.0*a);
11         q=sqrt(disc)/(2.0*a);
12         x1=p+q;
13         x2=p-q;//求出方程的两个根
14         printf("real roots:\nx1=%7.2f\nx2=%7.2f\n",x1,x2);//输出
15     }
16     return 0;
17 }
```

【例7.2】输入3个数 a , b , c , 要求按由小到大的顺序输出。

算法步骤

S1: if $a > b$, 将 a 和 b 对换
(交换后, a 是 a 、 b 中的小者)

S2: if $a > c$, 将 a 和 c 对换
(交换后, a 是 a 、 c 中的小者, 因此 a 是三者中最小者)

S3: if $b > c$, 将 b 和 c 对换
(交换后, b 是 b 、 c 中的小者, 也是三者中次小者)

S4: 顺序输出 a , b , c

【例7.2】输入3个数a , b , c , 要求按由小到大的顺序输出。

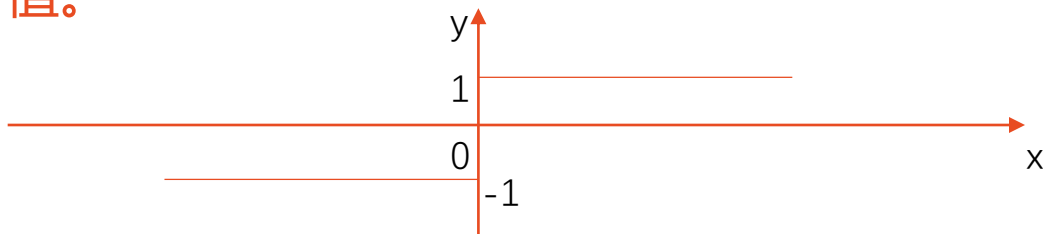
```
1 #include <stdio.h>
2 int main() {
3     float a,b,c,t;
4     scanf("%f%f%f",&a,&b,&c);
5     if (a>b) {
6         t=a; //借助变量t, 实现变量a和变量b互换值
7         a=b;
8         b=t;
9     } //互换后, a小于或等于b
10    if (a>c) {
11        t=a; //借助变量t, 实现变量a和变量c互换值
12        a=c;
13        c=t;
14    } //互换后, a小于或等于c
15    if (b>c) {
16        t=b; //借助变量t, 实现变量b和变量c互换值
17        b=c;
18        c=t;
19    } //互换后, b小于或等于c
20    printf("%5.2f,%5.2f,%5.2f\n",a,b,c); //顺序输出a,b,c的值
21    return 0;
22 }
```



在经过第1次互换值后, $a \leq b$, 经过第2次互换值后 $a \leq c$, 这样a已是三者中最小的(或最小者之一), 但是b和c谁大还未解决, 还需要进行比较和互换。经过第3次互换值后, $a \leq b \leq c$ 。

【例7.3】有一阶跃函数： $y = \begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$ 编一程序,输入一个x值(整数),

要求输出相应的y值。



算法步骤

先后用3个独立的if语句处理

S1: 输入x

S2: 若 $x < 0$,则 $y = -1$

S3: 若 $x = 0$,则 $y = 0$

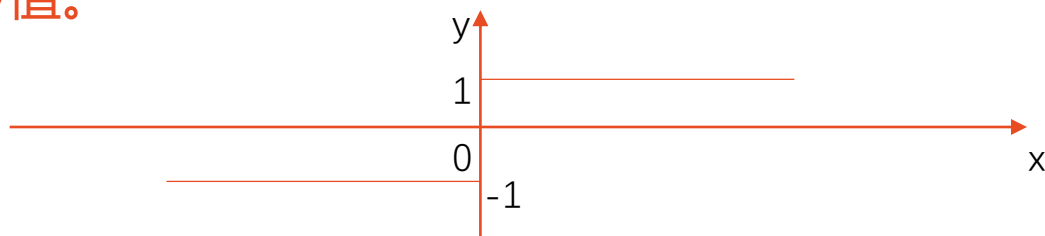
S4: 若 $x > 0$,则 $y = 1$

S5: 输出y

```
1 #include <stdio.h>
2 int main() {
3     int x,y;
4     scanf("%d",&x);
5     if (x<0)
6         y=-1;
7     if (x==0)
8         y=0;
9     if (x>0)
10        y=1;
11    printf("x=%d,y=%d\n",x,y);
12    return 0;
13 }
```


【例7.3】有一阶跃函数： $y = \begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$ 编一程序,输入一个x值(整数),

要求输出相应的y值。



算法步骤

用一个嵌套的if语句处理

S1: 输入x

S2: 若 $x < 0$,则 $y = -1$

S3: 否则

S4: 若 $x = 0$,则 $y = 0$

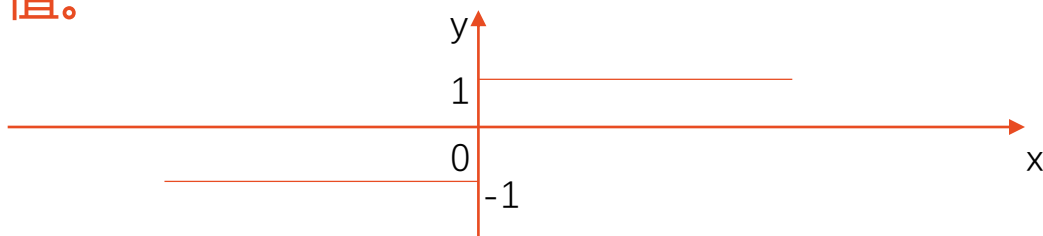
S5: 否则(即 $x > 0$),则 $y = 1$

S6: 输出y

```
1 #include <stdio.h>
2 int main() {
3     int x,y;
4     scanf("%d",&x);
5     if (x<0)
6         y=-1;
7     else {
8         if (x==0)
9             y=0;
10        else
11            y=1;
12    }
13    printf("x=%d,y=%d\n",x,y);
14    return 0;
15 }
```

【例7.3】有一阶跃函数： $y = \begin{cases} -1 & (x < 0) \\ 0 & (x = 0) \\ 1 & (x > 0) \end{cases}$ 编一程序,输入一个x值(整数),

要求输出相应的y值。



算法步骤

用一个嵌套的if语句处理

S1: 输入x

S2: 若 $x < 0$,则 $y = -1$

S3: 否则若 $x = 0$,则 $y = 0$

S4: 否则(即 $x > 0$),则 $y = 1$

S5: 输出y

```
1 #include <stdio.h>
2 int main() {
3     int x,y;
4     scanf("%d",&x);
5     if (x<0)
6         y=-1;
7     else if (x==0)
8         y=0;
9     else
10        y=1;
11    printf("x=%d,y=%d\n",x,y);
12    return 0;
13 }
```

【例7.4】打印成绩，若成绩 ≥ 90 分，输出“Very good”；若 $80 \leq \text{成绩} < 90$ 分，输出“Good”；若 $60 \leq \text{成绩} < 80$ 分，输出“Pass”；若成绩 < 60 分，否则输出“Fail”。

```
1 #include <stdio.h>
2 int main() {
3     float score;
4     printf("please input the score\n");
5     scanf("%f", &score);
6     if (score < 0)
7         printf("input error\n");
8     else if (score < 60)
9         printf("Fail\n");
10    else if (score < 80)
11        printf("Pass\n");
12    else if (score < 90)
13        printf("Good\n");
14    else if (score <= 100)
15        printf("Very Good\n");
16    else
17        printf("input error\n");
18    return 0;
19 }
```

多分支选择结构--开关语句switch

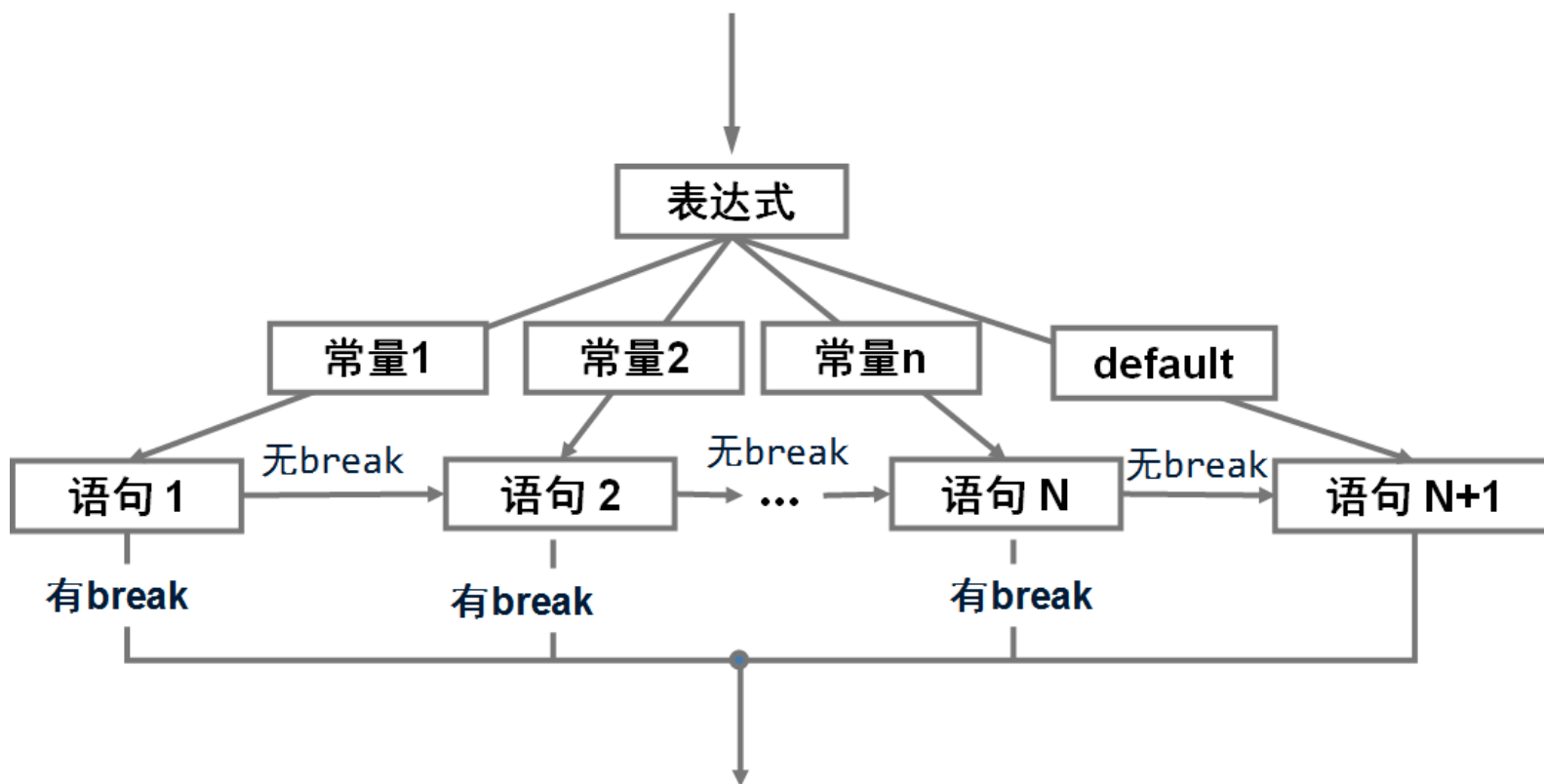
```
switch (表达式)
{
    case 常量1 : 语句系列1; [break;]
    case 常量2 : 语句系列2; [break;]
    .....
    case 常量n : 语句系列N; [break;]
    [default: 语句系列N+1;]
}
```

中断语句
可缺省

此处break语句的作用：跳出switch语句

default（缺省、默认）—— switch语句中表示未列出的情形

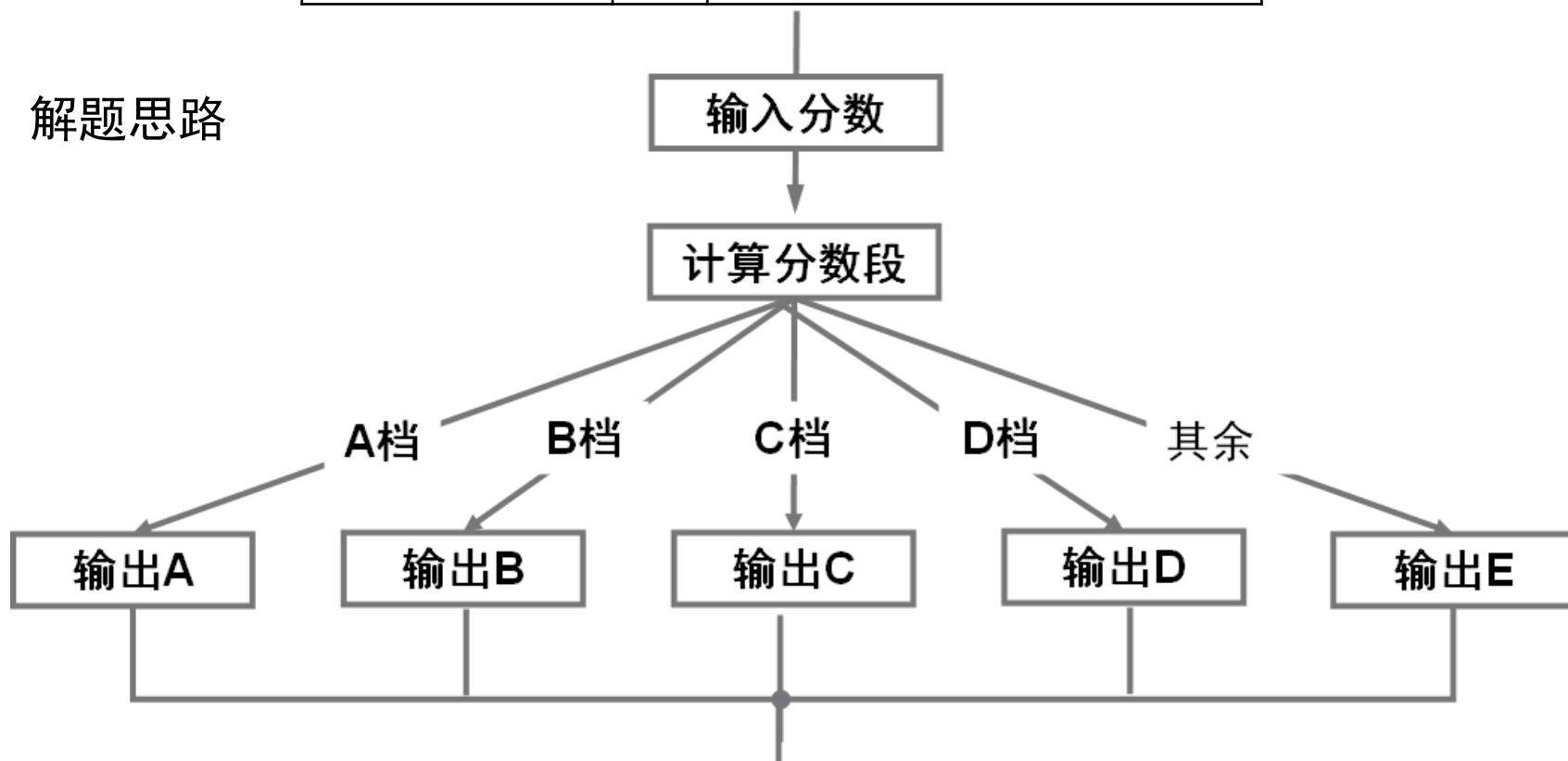
多分支选择结构



【例7.6】输入一个整数的百分制成绩score，转换成相应的五分制成绩grade并输出。

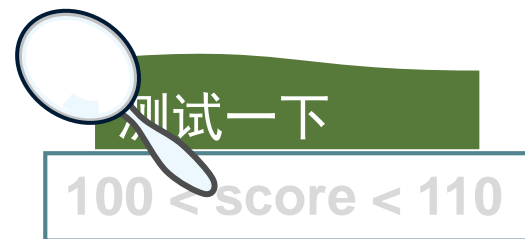
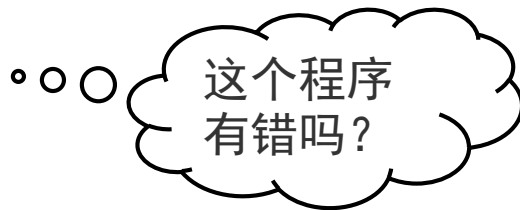
grade=	A	$90 \leq \text{score} \leq 100$
	B	$80 \leq \text{score} < 90$
	C	$70 \leq \text{score} < 80$
	D	$60 \leq \text{score} < 70$
	E	$\text{score} < 60$

解题思路



【例7.6】输入一个整数的百分制成绩score，转换成相应的五分制成绩grade并输出。

```
int    score;
scanf("%d", &score);
switch (score/10) {
case 10:
case 9:
    printf("%d-----A\n", score);    break;
case 8:
    printf("%d-----B\n", score);    break;
case 7:
    printf("%d-----C\n", score);    break;
case 6:
    printf("%d-----D\n", score);    break;
case 5:
case 4:
case 3:
case 2:
case 1:
case 0:
    printf("%d-----E\n", score);    break;
default:
    printf("Input  error\n");
}
```



【例7.6】输入一个整数的百分制成绩score，转换成相应的五分制成绩grade并输出。

改进后的程序：

```
1 #include <stdio.h>
2 int main() {
3     int score;
4     scanf("%d", &score);
5     if (score > 100 || score < 0) {
6         printf("input error\n");
7         return 0;
8     }
9     switch ( score / 10 ) {
10         case 10:
11         case 9:
12             printf("%d-----A\n", score);
13             break;
14         case 8:
15             printf("%d-----B\n", score);
16             break;
```

grade	A	$90 \leq \text{score} \leq 100$
	B	$80 \leq \text{score} < 90$
	C	$70 \leq \text{score} < 80$
	D	$60 \leq \text{score} < 70$
	E	$\text{score} < 60$

【例7.6】输入一个整数的百分制成绩score，转换成相应的五分制成绩grade并输出。

改进后的程序：

```
17         case 7:
18             printf("%d-----C\n", score);
19             break;
20         case 6:
21             printf("%d-----D\n", score);
22             break;
23         case 5:
24         case 4:
25         case 3:
26         case 2:
27         case 1:
28         case 0:
29             printf("%d-----E\n", score);
30             break;
31     }
32 }
```

【例7.7】设计程序，完成用户可以通过键盘输入数值，进行加、减、乘、除的运算。

数据分析	输入			输出
	float	char	float	float
'+'	a	+	b	a+b
'-'	a	-	b	a-b
'*'	a	*	b	a*b
'/'	a	/	b	a/b

以运算符做
switch中的
case值

【例7.7】设计程序，完成用户可以通过键盘输入数值，进行加、减、乘、除的运算。

```
1 #include <stdio.h>
2 int main() {
3     float a, b;
4     char c;
5     printf("please input an expression:a+(-,*,/)b\n");
6     scanf("%f%c%f", &a, &c, &b);
7     switch (c) {
8         case '+':
9             printf("%.2f\n", a + b);
10            break;
11        case '-':
12            printf("%.2f\n", a - b);
13            break;
14        case '*':
15            printf("%.2f\n", a * b);
16            break;
```

【例7.7】设计程序，完成用户可以通过键盘输入数值，进行加、减、乘、除的运算。

```
17         case '/':
18             if (b != 0) {
19                 printf("%.2f\n", a / b);
20                 break;
21             }
22             else {
23                 printf("error:divided by zero\n");
24                 break;
25             }
26         default:
27             printf("input error\n");
28     }
29     return 0;
30 }
```