



程序设计基础 (C语言)

郎大鹏

第4章 选择结构程序设计

- 4.1 if语句的基本形式
- 4.2 表达式与if语句条件判断
- 4.3 多分支选择结构及if嵌套
- 4.4 编程实例



第4章 选择结构程序设计

结构化程序设计是软件设计的第三次革命。结构化程序设计的基础是采用三种程序的控制结构。

1966年, BOHM & Jacopini 证明: 只要三种控制结构就能表达用一个入口和一个出口框图所能表达的任何程序逻辑。

三种控制结构如下:

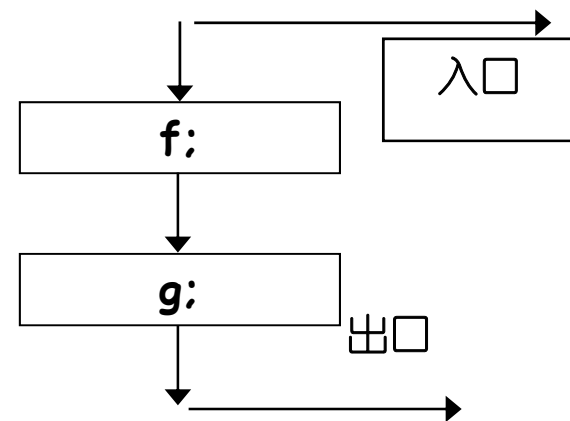
1. 顺序结构 Sequence

算法描述:

f;

g;

框图:

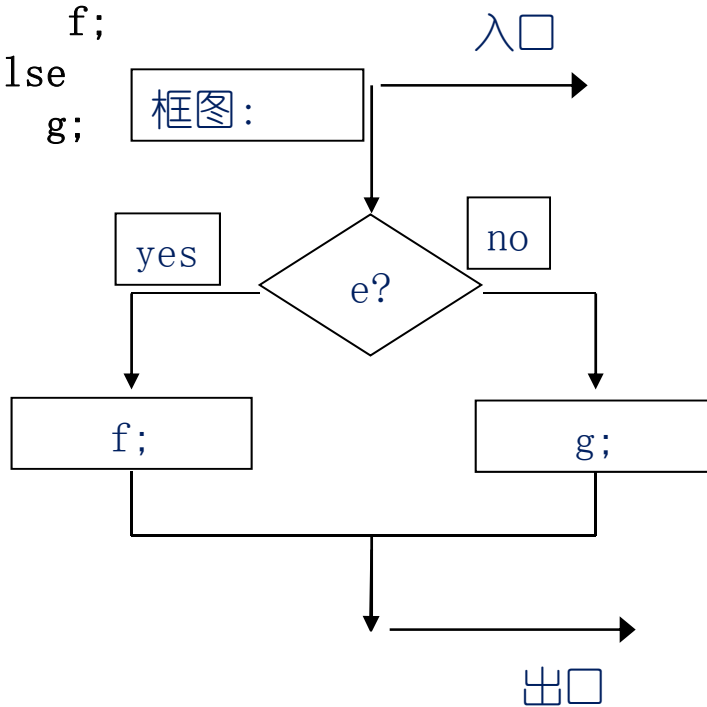


第4章 选择结构程序设计

2. 选择结构 Selection

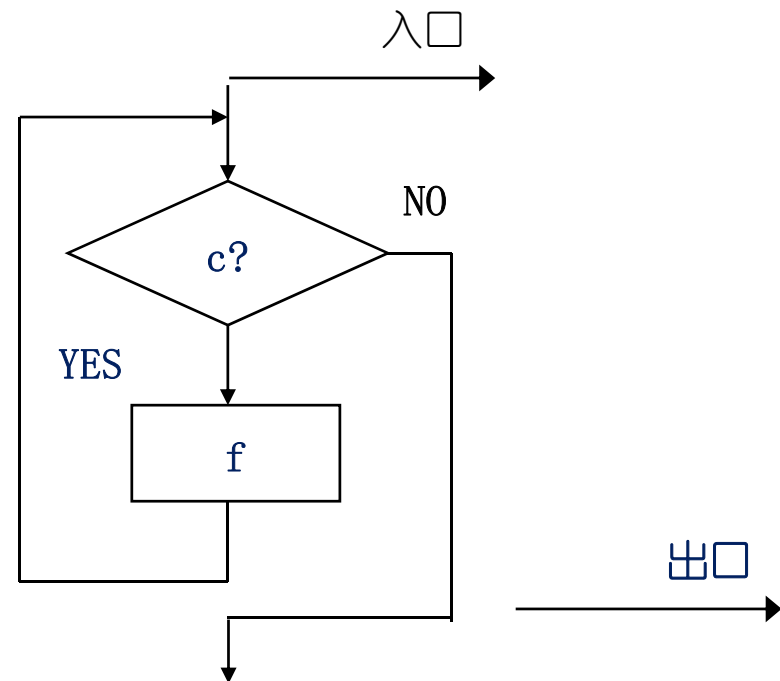
算法描述：

```
if(e)  
    f;  
else  
    g;
```



循环结构

```
while c is true  
do f
```



第4章 选择结构程序设计

结构化程序设计的核心思想

- 采用顺序、选择和循环三种基本结构作为程序设计的基本单元
 - 只有一个入口；
 - 只有一个出口；
 - 无死语句，即不存在永远都执行不到的语句；
 - 无死循环，即不存在永远都执行不完的循环。
- 采用“自顶向下、逐步求精”和模块化的方法进行结构化程序设计



第4章 选择结构程序设计

- 4.1 if语句的基本形式
- 4.2 表达式与if语句条件判断
- 4.3 多分支选择结构及if嵌套
- 4.4 编程实例



第4章 选择结构程序设计

• 4.1 if语句的基本形式

- 选择结构的一种最常用形式- 双分支结构

```
if (表达式)    语句1;  
              语句2;
```

表达式值“成立”，执行语句1;

表达式值“不成立”，执行语句2;



第4章 选择结构程序设计

• 4.1 if语句的基本形式

```
scanf("%d, %d",&a,&b);
```

```
if(a>b){ t=a; a=b; b=t; }
```

```
/*if语句的分支操作是复合语句*/
```

```
printf("%d, %d\n",a,b)
```

说明

if语句执行时，首先要计算if后圆括号中表达式等运算量的结果，若结果为“真”（非0），则执行if后的分支操作“语句”，然后继续下一条语句；若结果为“假”（0），则直接执行if语句的下一条语句。



第4章 选择结构程序设计^{图4.4}

• 4.1 if语句的基本形式

例4.1 当 $x \geq 10$ 时, $y = 3x - 11$, 求函数 y 的值, 试编写程序。

```
# include <stdio.h>
```

```
void main()
```

```
{ int x,y;
```

```
/*定义整型变量x、y
```

```
scanf("%d",&x);
```

```
/*从键盘输入整数赋值x变量*/
```

```
if(x>=10)
```

```
{y=3*x-11;
```

```
/*分支操作为复合语句*/
```

```
printf("%4d",y);
```

```
/*x>=10为“真”时, 赋值y并输出y */
```

```
}
```

```
/*如果x<10, 无输出, 直接结束 */
```

```
}
```



第4章 选择结构程序设计

• 4.1 if语句的基本形式

- 选择结构的一种最常用形式- 双分支结构

if (表达式)

 语句1;

else

 语句2;

语句3

- 表达式值非0时，执行语句1，然后语句3;
表达式值为0时，执行语句2，然后语句3;



第4章 选择结构程序设计

• 4.1 if语句的基本形式

■ else部分可以没有。

if (表达式)

语句1;

语句3

当表达式值为0时，直接执行语句3

if-else嵌套使用时，注意else和谁配对的问题



第4章 选择结构程序设计

• 4.1 if语句的基本形式

例如：

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

说明

if语句执行时，首先要计算if后圆括号中表达式的值，如果表达式结果为“真”（非零），则执行if后“语句1”，然后去执行if的下一条语句；如果表达式结果为“假”（零），则执行else后“语句2”，然后去执行if的下一条语句。



第4章 选择结构程序设计

• 4.1 if语句的基本形式

例4.5 写出下面程序的输出结果

```
#include <stdio.h>

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

程序运行后，如果从键盘上输入5，则输出结果是4。



第4章 选择结构程序设计

- 4.1 if语句的基本形式
- 4.2 表达式与if语句条件判断
- 4.3 多分支选择结构及if嵌套
- 4.4 编程实例



第4章 选择结构程序设计

• 4.2 表达式与if语句条件判断

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

if后表达式并不只限于关系表达式或逻辑表达式，可以是任意运算对象，C语言规定其运算结果为非“0”时，即为“真”，只有其运算结果为“0”时，才为“假”。



第4章 选择结构程序设计

• 4.2 表达式与if语句条件判断

4.2.1 关系运算和逻辑运算

1. 关系运算

关系运算是比较两个表达式值的数值相互关系的运算。

运算符	运算关系	实例
>	大于	$a > b$
>=	大于等于	$a \geq b$
<	小于	$2 < 1$
<=	小于等于	$c \leq d$
==	等于	$1 == c$
!=	不等于	$1 != 3$

关系运算规则：参加运算的表达式从左到右按关系运算符提供的关系进行比较，满足关系得到整型值1，不满足关系得到整型值0。

```
int a=1, b=3, c, d;  
c=a>b ;/*c的值为0*/  
d=a+2<=b+3; /*d的值为1*/
```



第4章 选择结构程序设计

• 4.2 表达式与if语句条件判断

2. 逻辑运算

运算符	逻辑关系	举例
&&	逻辑与	a>2&&a<3
 	逻辑或	s<2 s>6
!	逻辑非	!a

其中，！运算是单目运算。

逻辑运算运算时判断对象真假的运算。运算对象非**0**代表逻辑真，是**0**代表逻辑假。也就是说任何类型的量都有逻辑值。逻辑运算得到整型值。运算结果为真时，得到整型值**1**，运算结果为假时，得到整型值**0**。



第4章 选择结构程序设计

• 4.2 表达式与if语句条件判断

练习：求下列关系表达式的值。

- ① $x \neq y$
- ② $a \% 2 == a$
- ③ $p > 'm'$
- ④ $c > a < b$
- ⑤ $a != b <= c > 3$
- ⑥ $p + 1 == 'e' != a$



第4章 选择结构程序设计

• 4.2 表达式与if语句条件判断

if后的圆括号内，可以是任意运算量和任意表达式，系统最终以0和非0来判定它们属于“真”或“假”，其括号内的结果必定是逻辑量“真”或“假”。

例如：

if(56)、if(-1)、if(a=8)、if('a')：条件判断都为真。

if(!-5)、if(8-8)、if(a=0)、if(0&&a&&b+9<=a++)：条件判断都为假。

if(a==0)、if(a-8!=0)：条件判断结果要根据a的值而定。



第4章 选择结构程序设计

• 4.2 表达式与if语句条件判断

例4.10 输入3个整数num1、num2、num3，求最大值。

算法分析：

(1) 任取3个整数中的一个赋值给max变量，如： $\text{max}=\text{num1}$ ；

(2) 用其余的整数依次与max比较：

如： $\text{num2} > \text{max}$ ，则 $\text{max} = \text{num2}$ ，比较完所有的数后，max中的数就是最大值。



第4章 选择结构程序设计

• 4.2 表达式与if语句条件判断

```
#include "stdio.h"

void main()
{
    int num1, num2, num3, max;
    printf (" Please input three numbers: ");
    scanf ("%d, %d , %d", &num1, &num2, &num3);
    max = num1;
    if (num2 > max) max = num2;          /* max是num1与 num2中最大的*/
    if (num3 > max) max = num3;          /* max是num1、 num2、 num3中最大的*/
    printf ("The three numbers are:%d, %d , %d \n", num1, num2, num3);
    printf ("max=%d\n ", max);
}
```



2013年2月1日

第4章 选择结构程序设计

- 4.1 if语句的基本形式
- 4.2 表达式与if语句条件判断
- 4.3 多分支选择结构及if嵌套
- 4.4 编程实例



第4章 选择结构程序设计

• 4.3 多分支选择结构及if嵌套

举例

学生成绩分类：90分以上为‘A’等，80-89分为‘B’等，70-79分为‘C’等……；个人所得税：按收入分档纳税；人口统计：按年龄分为老、中、青、少、儿童；工资统计分类、银行存款分类等等，根据不同的情况，需要选择不同的操作方法。

这种情况应当如何处理呢？



第4章 选择结构程序设计

• 4.3 多分支选择结构及if嵌套

1. 语句格式

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n;



第4章 选择结构程序设计

• 4.3 多分支选择结构及if嵌套

例4.12 :

```
if(score >=90) grade='A';  
    else if(score >=80) grade='B';  
        else if(score >=70) grade='C';  
            else if(score >=60) grade='D';  
                else grade='E';  
printf( "grade is %c\n", grade);
```



第4章 选择结构程序设计

• 4.3 多分支选择结构及if嵌套

例4.14 输入一个整数与预定的数比较，并给出“大”或“小”或“相等”的提示。

```
#include "stdio.h"
void main( )
{ int i, number=200;
  scanf("%d",&i);
  if(i>number)
    printf("It's large!\n");
  else if (i<number) printf("It's small!\n");
  else printf("It's right!\n");
}
```



第4章 选择结构程序设计

• 4.3 多分支选择结构及if嵌套

思考：此程序中的所有else是否可以省略，即：用单分支选择结构是否可以实现？

如果if多分支结构的if语句层数较多，则条件判断也多，而且程序冗长，可读性降低。初学者要尽量避免使用这种方法解决分支较多的问题。c语言提供了switch语句处理多分支选择问题。



第4章 选择结构程序设计

• 4.3.2 switch语句

语句格式

```
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;
}
```



第4章 选择结构程序设计

• 4.3.2 switch语句

如何用switch语句来实现例4.13？

例4.13 从键盘输入1个成绩：0-100的整数，根据成绩输出等级。

(90-100: A 、 80-89: B 、 70-79: C 、 60-69: **D** 、 0-59: **E**)



第4章 选择结构程序设计

• 4.3.2 switch语句

```
#include "stdio.h"
void main( )
{ int score, grade;    printf ("score= ");    scanf ( "%d" , &score );
  grade= score/10;    /*将成绩整除10，转化成case标号*/
  switch (grade) {
    case 10:
    case 9: printf ("A\n"); break;
    case 8: printf ("B\n"); break;
    case 7: printf ("C\n"); break;
    case 6: printf ("D\n"); break;
    default: printf ("E\n");
  }
```



第4章 选择结构程序设计

• 4.3.2 switch语句

例4.15 计算器功能程序。用户输入数和四则运算符，输出其计算结果。

算法分析：

此题的关键在于运算符的选择，用switch语句实现，switch后面括号中表达式应是字符型，那么case后常量也应该是字符。



第4章 选择结构程序设计

• 4.3.2 switch语句

```
#include "stdio.h"
void main( )
{
    float x,y;
    char z;
    printf("input x+(-,*,/)y:");          /*屏幕显示输入提示*/
    scanf("%f%c%f",&x,&z,&y);
    switch(z)
    {
        case '+': printf("%f\n",x+y);break;
        case '-': printf("%f\n",x-y);break;
        case '*': printf("%f\n",x*y);break;
        case '/': printf("%f\n",x/y);break;
        default: printf("Input error!\n");
    }
}
```



第4章 选择结构程序设计

• 4.3.3 if嵌套结构

1. 语句格式:

形式1:

```
if ( )  
    if ( ) 语句1  
    else 语句
```

形式2:

```
if ( )  
    if ( ) 语句1  
    else 语句2  
else 语句3
```



第4章 选择结构程序设计

• 4.3.3 if嵌套结构

例：

比较两个数的大小关系。

```
void main()
{
    int a,b;
    printf("please input A,B:");
    scanf("%d%d",&a,&b);
    if(a!=b)
        if(a>b) printf("A>B\n");
        else printf("A<B\n");
    else printf("A=B\n");
}
```



第4章 选择结构程序设计

• 4.3.3 if嵌套结构

本例中用了if语句的嵌套结构。
采用嵌套结构实质上是为了进行
多分支选择，实际上有三种选择
即 $A > B$ 、 $A < B$ 或 $A = B$ 。

如下面程序，而且程序更加清晰。
因此，在一般情况下**较少**使用
if语句的嵌套结构，以使程序更
便于阅读理解。

```
void main()
{
    int a,b;
    printf("please input A,B:");
    scanf("%d%d",&a,&b);
    if(a==b) printf("A=B\n");
    else if(a>b) printf("A>B\n");
        else printf("A<B\n");
}
```



第4章 选择结构程序设计

• 4.3.3 if嵌套结构

嵌套结构中，需要注意：

if语句内嵌在if子句中时，为了避免视觉混淆，在if与else的个数不相同的情况下，尽量使用花括号解决if和else的对应问题。在程序书写时，也要注意位置上的对称和相应的层次感，养成良好的程序书写习惯也是很重要的。



第4章 选择结构程序设计

• 4.3.3 if嵌套结构

嵌套结构中，需要注意：

例A:

```
if ( )  
    if ( )  
        语句1
```

```
else 语句2
```

例B:

```
if ( )  
    { if ( )  
        语句1 }
```

/*复合句作为第一个if的子句*/

```
else 语句2    /* else与外层if匹配*/
```



```
if ( )
```

/* 是 没有else的if结构*/

```
{ if ( ) 语句1
```

```
    else 语句2 }
```

/* else与内层if匹配*/



第4章 选择结构程序设计

• 4.3.3 if嵌套结构

例4.19 设计一函数程序，输入一个x值，输出y值.

$$y = \begin{cases} x & x < 1 \\ 2x-1 & 1 \leq x < 10 \\ 3x-11 & x \geq 10 \end{cases}$$

分析：可以用if单分支结构，也可以用if语句嵌套结构完成，但两种方法有差异。



第4章 选择结构程序设计

• 4.3.3 if嵌套结构

(1)用if单分支结构实现:

```
#include "stdio.h"

void main()
{ float x,y;
  printf("Input x:");
  scanf("%f", &x);
  if(x<1) y=x;
  if(x>=1&& x<10) y=2*x-1;
  if(x>=10) y=3*x-11;
  printf("y=%.2f\n", y);
}
```

此程序不论x为何值，三个if语句都要被执行。



2013年2月1日

(2)用if语句嵌套结构实现: 算法如图4.24

```
#include "stdio.h"

void main()
{ float x,y;
  printf("Input x:");
  scanf("%f", &x);
  if(x>=1) /* 不可加分号，语句没有i结束*/
    if(x>=10) y=3*x-11;
    else y=2*x-1;
    /*为第一个if的else，是x<1的情况*/
  printf("y=%.2f\n", y);
}
```

第4章 选择结构程序设计

- 4.1 if语句的基本形式
- 4.2 表达式与if语句条件判断
- 4.3 顺多分支选择结构及if嵌套
- 4.4 编程实例



第4章 选择结构程序设计

• 4.4 编程实例

例4.22 写出下面各逻辑表达式的值，设 $a=3$ ， $b=4$ ， $c=5$ 。

(1) $a+b>c \& \&b==c$

(2) $a||b+c \& \&b-c$

(3) $!(a>b) \& \&!c || 4$

(4) $!(x=a) \& \&(y-b) \& \&0$

解答：(1) $3+4>5 \& \&4==5$ ， $7>5 \& \&4==5$ ， $1 \& \&0$ ：0

(2) $3||4+5 \& \&4-5$ ， $3||9 \& \&-1$ ， $3||1$ ：1

(3) $!(3>4) \& \&!5 || 0$ ， $!0 \& \&0 || 0$ ， $1 \& \&0 || 0$ ：0

(4) $!(x=3) \& \&(y-4) \& \&4$ ， $!3=0$ ， $0 \& \&...$ ：0



第4章 选择结构程序设计

• 4.4 编程实例

例**4.23** 输入4个整数，要求从小到大顺序输出。

分析：用依次比较的方法，通过中间变量交换较大、较小数。两个数比较大小需要比较1次，3个数要比较3次，4个数要比较6次等。



第4章 选择结构程序设计

• 4.4 编程实例

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



第4章 选择结构程序设计

• 4.4 编程实例

例4.24 输入一个不多于5位的正整数，求它是几位数并输出每一位数字。

分析：输入5位之内正整数n，

if(n>9999) c=5; (5位数);

else if(n>999) c=4; (4位数)

.....;

对于输出每一位数字，万位= $n/10000$ 取整数，千位= $(n-\text{万位} \times 10000)/1000$。



第4章 选择结构程序设计

• 4.4 编程实例

```
#include <stdio.h>
#include "math.h"
void main()
{ long n;
  int g,s,b,q,w,c;
  printf("input anumber(<100000): ");
  scanf("%ld",&n);
  if(n>9999) c=5;
  else if(n>999) c=4;
    else if(n>99) c=3;
      else if(n>9) c=2;
        else c=1;
  printf("count=%d\n",c);
```



```
printf("digit: ");
  w=n/10000;
    /* w为万位数字*/
  q=(int)(n-w*10000)/1000;
    /* q为千位千数字*/
  b=(int)(n-w*10000-q*1000)/100;
    /* b为百位数字*/
  s=(int)(n-w*10000-q*1000-b*100)/10;
    /* s为十位数字*/
  g=n%10;
    /* g为个位数字*/
  printf("%d, %d, %d, %d, %d\n",w,q,b,s,g);
}
```

第4章 选择结构程序设计

- 4.4 编程实例

例4.26 输入星期几（1—7），输出其对应的英文单词。

分析：用**switch**结构比较好，
注意使用**break**语句。



第4章 选择结构程序设计

• 4.4 编程实例

```
#include "stdio.h"
void main()
switch(a)
{ case 1: printf("Monday\n"); break;
  case 2: printf("Tuesday\n"); break;
  case 3: printf("Wednesday\n"); break;
  case 4: printf("Thursday\n"); break;
  case 5: printf("Friday\n"); break;
  case 6: printf("Saturday\n"); break;
  case 7: printf("Sunday\n"); break;
  default: printf("Error!\n");
}scanf("%d",&a);
}
```



第4章 选择结构程序设计

例4.24 输入一个不多于5位的正整数，求它是几位数并输出每一位数字。

算法分析：

输入5位之内正整数n，if(n>9999) c=5; (5位数);

else if(n>999) c=4; (4位数)

.....;



第4章 选择结构程序设计

于输出每一位数字，万位= $n/10000$ 取整数，千位
= $(n - \text{万位} * 10000) / 1000$。

程序如下：

```
#include <stdio.h>
#include "math.h"
void main()
{ long n;
  int  g,s,b,q,w,c;
  printf("input anumber(<100000): ");
  scanf("%ld",&n);
  if(n>9999) c=5;
  else if(n>999) c=4;
  else if(n>99) c=3;
  else if(n>9) c=2;
  else c=1;
```

```
    printf("count=%d\n",c);
    printf("digit: ");
    w=n/10000;    /* w为万位数字*/
    q=(int)(n-w*10000)/1000; /* q为千位千数字*/
    b=(int)(n-w*10000-q*1000)/100;
    /* b为百位数字*/
    s=(int)(n-w*10000-q*1000-b*100)/10;
    /* s为十位数字*/
    g=n%10;    /* g为个位数字*/
    printf("%d, %d, %d, %d, %d\n",w,q,b,s,g);
}
```



第4章 选择结构程序设计

例4.25 求一元二次方程 $ax^2+bx+c=0$ 的根。

算法分析：一元二次方程的根有下列情况：

当 $a=0$ ， $b=0$ 时，方程无解；

当 $a=0$ ， $b\neq 0$ 时，方程只有一个实根 $-c/b$ ；

当 $a\neq 0$ 时，方程的根如下：

当 $b^2-4ac\geq 0$ 时有两个实根；

当 $b^2-4ac< 0$ 时有两个虚根。



```

#include "stdio.h"
#include "math.h"
void main()
{ float a,b,c,disc,x1,x2,p,q;
  printf("a,b,c=");
  scanf("%f,%f,%f",&a,&b,&c);
  if(fabs(a)<=1e-6 && fabs(b)<=1e-6)
      printf("havn'troot!\n");
  else if (fabs(a)<=1e-6 && fabs(b)>1e-6)
      printf("single root: %8.4f\n",-c/b);
  else
  { disc=b*b-4*a*c;
    if (fabs(disc)<=1e-6||disc>1e-6)
    { x1=(-b+sqrt(disc))/(2*a);
      x2=(-b-sqrt(disc))/(2*a);
      printf("two root!\n");
      printf("x1=%8.4f\n",x1);
      printf("x2=%8.4f\n",x2); }
    else { p=-b/(2*a); q=sqrt(-disc)/(2*a);
      printf("two imaginary root !\n");
      printf("x1=%8.4f+%8.4fi\n",p,q);
      printf("x2=%8.4f-%8.4fi\n",p,q); }
  }
}

```

/* 取实数绝对值数学函数: `fabs()` */

/*a=0, b=0, 方程无解 */

/*a=0, b≠0, 方程只有一个实根-c/b*/

/*当 $b^2 - 4ac \geq 0$ 时有两个实根*/

/*当 $b^2 - 4ac < 0$ 时有两个虚根*/

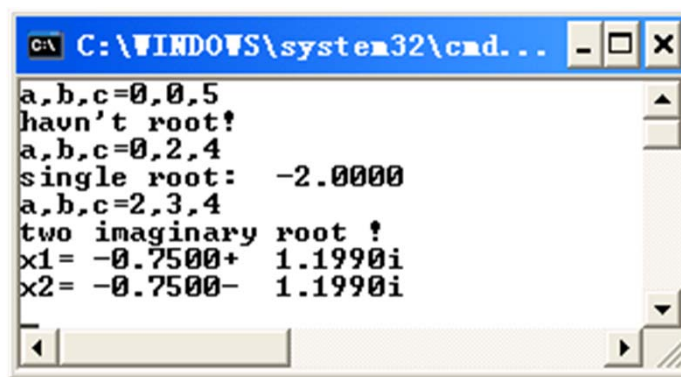


第4章 选择结构程序设计

注意:

(1) $disc$ 代表 b^2-4ac ，先计算 $disc$ 的值，以减少以后的重复计算。 p 代表实部， q 代表虚部，以增加可读性。

(2) 由于 $a, b, disc$ (即 b^2-4ac) 为实数，而实数在计算和存储时会有一些微小的误差，因此不能直接进行如下判断：if ($a==0$)……。所以采取的办法是判别 a 的绝对值 $fabs(a)$ 是否小于一个很小的数（如 10^{-6} ），如果小于此数，就认为 $a=0$ 。



```
C:\WINDOWS\system32\cmd...  
a,b,c=0,0,5  
havn't root!  
a,b,c=0,2,4  
single root: -2.0000  
a,b,c=2,3,4  
two imaginary root !  
x1= -0.7500+ 1.1990i  
x2= -0.7500- 1.1990i
```



第4章 选择结构程序设计

例4.26 输入星期几（1—7），输出其对应的英文单词。

分析：用switch结构比较好，注意使用break语句。



第4章 选择结构程序设计

程序如下：

```
#include "stdio.h"

void main()
{   int a;

    printf("please input a week number:");
    scanf("%d",&a);
    switch(a)
    {   case 1:printf("Monday\n");break;
        case 2:printf("Tuesday\n"); break;
        case 3:printf("Wednesday\n"); break;
        case 4:printf("Thursday\n"); break;
        case 5:printf("Friday\n"); break;
        case 6:printf("Saturday\n"); break;
        case 7:printf("Sunday\n"); break;
        default:printf("Error!\n");
    }
}
```

