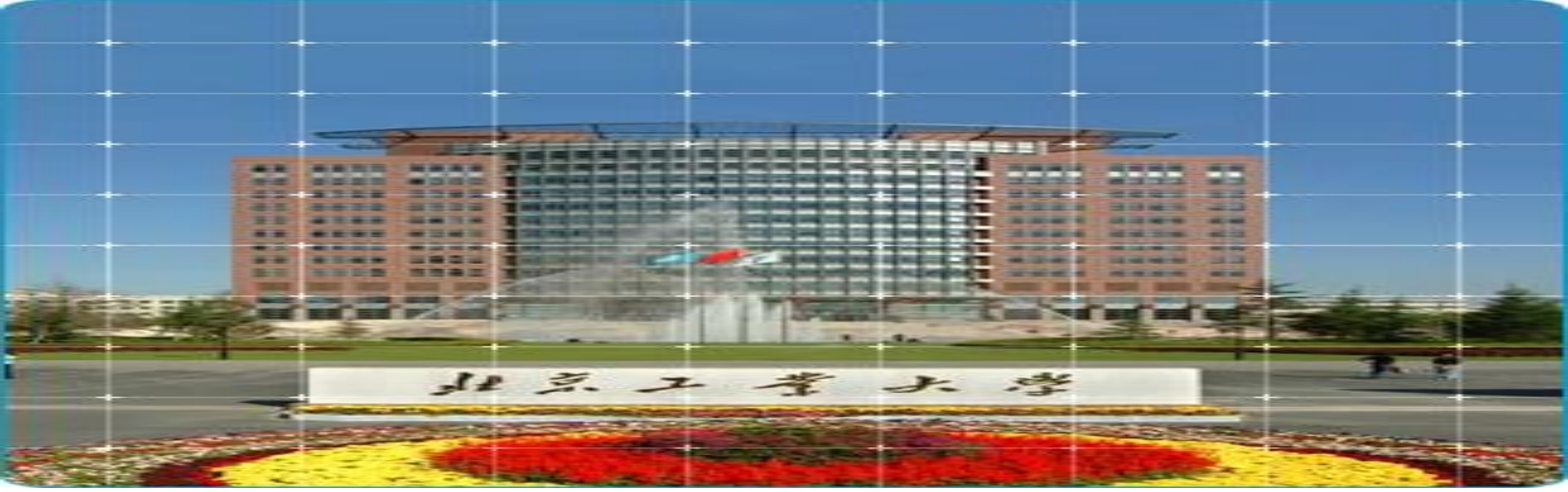


第2章 C语言程序设计基础

- 基本数据类型和输入输出
  - 输入生日并显示
  - 汇率换算
  - 字符加密
  - 考核通过了吗？
- 顺序结构
  - 计算圆的周长和面积
- 选择分支结构
  - 判断数字的奇偶性（if）
  - 今年是闰年吗？（if-else）
  - 求解一元二次方程（if-else if）
  - 四则运算（switch）
  - 计算存款利息（switch）
  - 可以构成三角形吗？（综合案例）

本章问题

- 怎样编写程序，在屏幕上显示一些信息？
- 怎样编写程序，实现简单的数据处理？
- 什么是分支结构？它的作用是什么？
- switch 语句中的 break 起什么作用？
- 数据在内存中是如何存储的？

2.1 导例：输入生日并显示

1. 问题描述

从键盘输入个人生日信息，并在屏幕上显示相应信息。

2. 问题分析

问题涉及到了输入和输出问题。

使用标准输入函数scanf( )接收键盘输入的信息，

使用输出函数printf( )屏幕输出。

从键盘输入的信息要想输出到屏幕上，需要先将数据存储到内存空间保存起来，然后对其输出。因此需要定义一个变量用于保存键盘输入的数。在这里可以定义两个整型变量分别用于保存生日的月份和日期信息。

3. 算法描述

定义int变量month和day；

用printf( )在屏幕上显示提示信息“Enter the month and the day of your birthday:”；

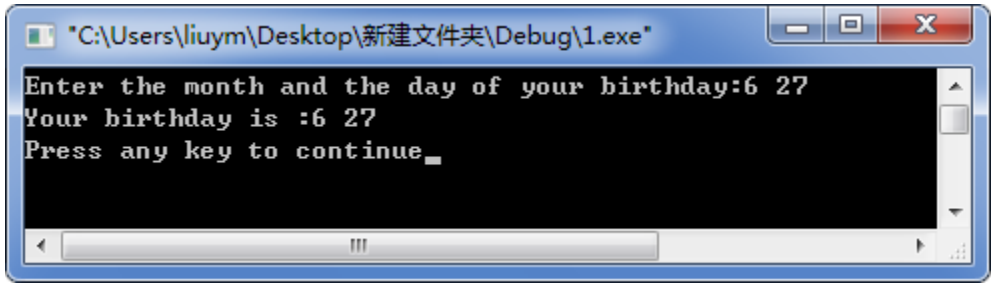
用scanf( )从键盘输入生日月份和日期，分别存储在变量month和day中；

用printf( )在屏幕上显示生日信息。

4. 程序实现

```
#include <stdio.h>              /*编译预处理命令*/
int main( )                     /*定义了一个名字为main的函数*/
{
    int month, day;             /*定义两个整形变量month和day*/
  printf("Enter the month and the day of your birthday:");
  scanf("%d%d", &month,&day);   /*接收键盘输入的两个整数*/
    printf("Your birthday is :%d %d\n", month, day);
    return 0;
}
```

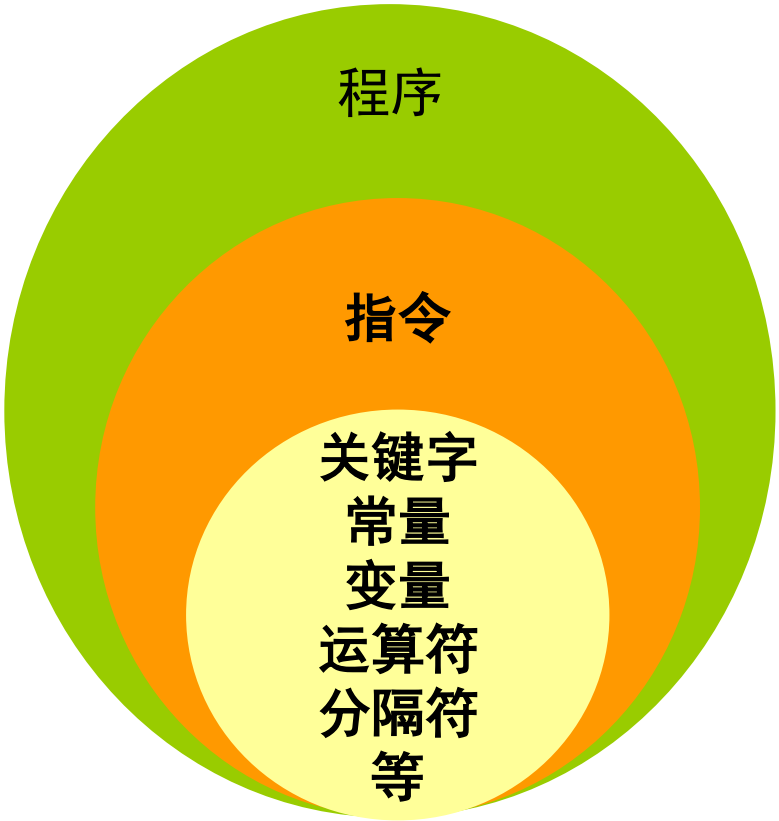
5. 运行结果



C语言程序的基本结构

预编译命令	**#include <stdio.h>**
函数类型 **main**（函数参数）	int main( )
{　函数体开始	{
声明部分	int month, day;
执行部分	printf("Enter the month and the day of your birthday:"); scanf("%d%d", &month,&day printf("Your birthday is :%d %d", month, day); return 0;
}　函数体结束	}

C语言中的基本元素



2.2基本数据类型与基本输入输出

数据类型	类型说明符	字节数	取值范围
字符型	char	1	C字符集
整型、长整型	int、long int	4	-2147483648~2147483647即-2^{31}~$(2^{31}-1)$
短整型	short int	2	-32768~32767即-2^{15}~$(2^{15}-1)$
无符号整型	unsigned int	4	0~4294967295即0~$(2^{32}-1)$
无符号短整型	Unsigned short int	2	0~65535即0~$(2^{16}-1)$
单精度型	float	4	负数：-3.4028235×10^{38}~-1.401298×10^{-45} 正数：1.401298×10^{-45}~3.4028235×10^{38}
双精度型	double	8	负数：$-1.79769313486231570\times10^{308}$~ $-4.94065645841246544\times10^{-324}$ 正数：$4.94065645841246544\times10^{-324}$~ $1.79769313486231570\times10^{308}$
长双精度型	Long double	16	10^{-4931} ~10^{4932}

关键字与标识符

- 关键字，是程序设计语言保留的特殊标识符，又称保留字。标准的C语言中规定了32个关键字，在C语言中不允许用户标识符与关键字相同。例如：int
- 标识符就是用户自己标识程序中某个对象的名字的，这些对象可以是常量、变量、数据类型、语句、函数等。

  常量在程序运行过程中始终不发生变化，变量则是其值可以改变的量。

变量的定义

变量名：小写字母；见名知义

变量定义的一般形式：

类型名　变量名表；

例如：

```
int month, day;      //定义整型变量
float x;             //定义单精度浮点型变量
double area, length;    //定义双精度浮点型变量
```

double型数据比float精度高，取值范围大

变量的定义

- 定义变量时要指定变量名和数据类型

  类型名　变量名表；

  int month, day;

  float x;

  double area, length;

- 变量名代表内存中的一个存储单元

  存放该变量的值

- 该存储单元的大小由变量的数据类型决定

- C语言中的变量代表保存数据的存储单元
- 数学中的变量代表未知数

  x = x+1

变量命名规则

- 变量的命名规则：
  - 变量名可以由字母、数字和 _（下划线）组合而成
  - 变量名不能包含除 _ 以外的任何特殊字符，如：%、# 、逗号、空格等
  - 变量名必须以字母或 _（下划线）开头
  - 变量名不能包含空白字符（换行符、空格和制表符称为空白字符）
  - C 语言中的某些词（例如 **int** 和 **float** 等）称为保留字，具有特殊意义，不能用作变量名
  - C 语言区分大小写，因此变量 **price** 与变量 **PRICE** 是两个不同的变量

变量命名

有效名称

principal

lastname

cost_price

city

marks_3

无效名称

123rate

zip code

currency$

discount%

变量的定义与使用

变量必须先定义，后使用。

```
#include <stdio.h>
int main( )
{
    float math, computer,total;
    printf("Enter your score:");
    scanf("%d%d", &math,&computer);
    total = math+ computer;
    printf(" total = %d\n", total);
    return 0;
}
```

应该先赋值，后引用

一个变量名只能定义一次
变量一般都定义在程序的头上
不能定义在程序的中间或后面

导例：汇率换算

1. 问题描述

假设人民币对美元的汇率是0.1642，计算用户输入的人民币值可以兑换多少美元。

2. 问题分析

人民币兑换美元的计算公式为：dollar=rmb×rate

3. 算法描述

确定汇率，rate=0.1642；

接收用户输入的人民币值，将其赋值给rmb；

计算公式：dollar = rmb *rate；

屏幕输出结果。

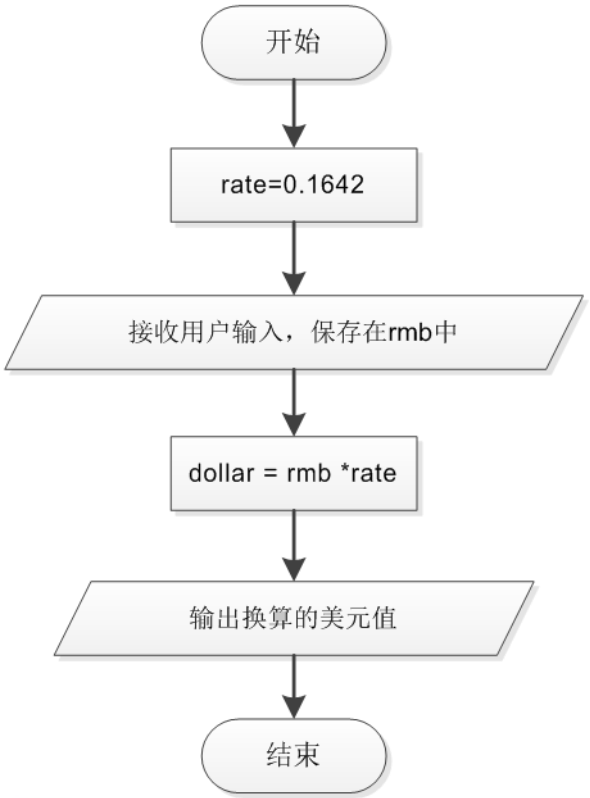
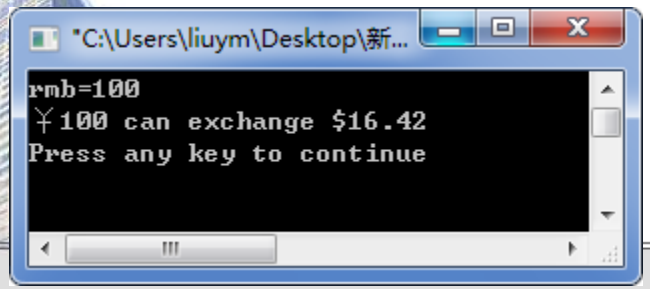


图2-1汇率换算流程图



4. 程序实现

```
#include <stdio.h>
 void main( )
{
   int  rmb;      /*定义整型变量，存放人民币值*/
   float  rate;    /*定义单双精度浮点型变量，存放汇率值*/
   float  dollar; /*定义单精度浮点型变量，存放美元值*/
   rate=0.1642;            /*对变量rate赋值*/
   printf("rmb=");
   scanf("%d",&rmb);  /*接收用户输入*/
   dollar=rmb*rate;       /*汇率换算*/
   printf(" ￥%d can exchange $%.2f\n",rmb,dollar);
    /*调用printf()函数输出结果*/
}
```

标准输入函数scanf()

- 函数scanf( )的一般调用格式为：

  scanf ("<格式控制字符串>",地址列表)

  格式说明符和地址列表在数量和类型上要一一对应。

  - int型 ： %d
  - float型： %f
  - double型： %lf

- 例如：

```
int month, day;
scanf("%d%d", &month,&day);
```

"%d%d"是格式控制字符串，%d和%f是格式说明符

&是地址符，& month表示变量month的存储地址。

注意：普通字符：原样输入　尽量不要出现普通字符

例如：

```
scanf("%lf", &x);
```
输入：9.5

```
scanf("%d,%d,%d",&x,&y,&z);
```
输入：9, 5, 7

```
scanf("%d;%d;%d",&x,&y,&z);
```
输入：9; 5; 7

```
scanf("%d-%d-%d",&x,&y,&z);
```
输入：9−5−7

```
scanf("x=%d y=%d z=%d",&x,&y,&z);
```
输入：x=9 y=5 z=7

" "内最好不要加任何符号，如果要加提示，这些提示不会自动显示出来，而必须由操作者手动输入，否则出现数据错误。

可以采用输入提示的输入方式：

```
printf("x="); scanf("%d",&x);
printf("y="); scanf("%d",&y);
printf("z="); scanf("%d",&z);
```

```
x=9
y=5
z=7
```

标准输出函数printf()

- 函数printf( )的一般调用格式为：

  printf("<格式控制字符串>",输出列表)

  在函数printf()中的格式控制字符串中还可以包含普通字符，在显示中起到提示作用。

- 例如：

  printf(" Your birthday is :%d %d\n", month, day);

- printf()的输出参数也要和格式说明符一一对应。
  - int型 ： %d
  - float double型： %f

- printf()函数有一些附加格式说明符，可用于修饰格式说明符，格式为：%m和%m.n。
- 如

"%.2f"中的".2"表示输出2位小数；

"%.2s"，则表示只输出字符串的前两个字符；

"%-10.2f"表示该变量一共占10位，包括整数部分、小数点和小数点后位数，且小数点后只能保留2位，“-”表示采用左对齐方式。

表2-5 常用的格式说明符

格式说明符	功能说明
%d	输入或输出十进制整数
%o	输入或输出八进制整数
%x	输入或输出十六进制整数
%f	输入或输出实数
%c	输入或输出单个字符
%s	输入或输出字符串

导例：字符加密

1. 问题描述

输入字符，为其加密。

2. 问题分析

加密是保证信息安全的常用方法。选用一种最简单的方式，就是字符变换，变换方式为：字母转换为其后第三个字母，a变为d，b变为e，c变为f……。

3. 算法描述

- 接收用户输入的一个字符，将其赋值给ch；
- 进行字符变换；
- 屏幕输出加密结果.

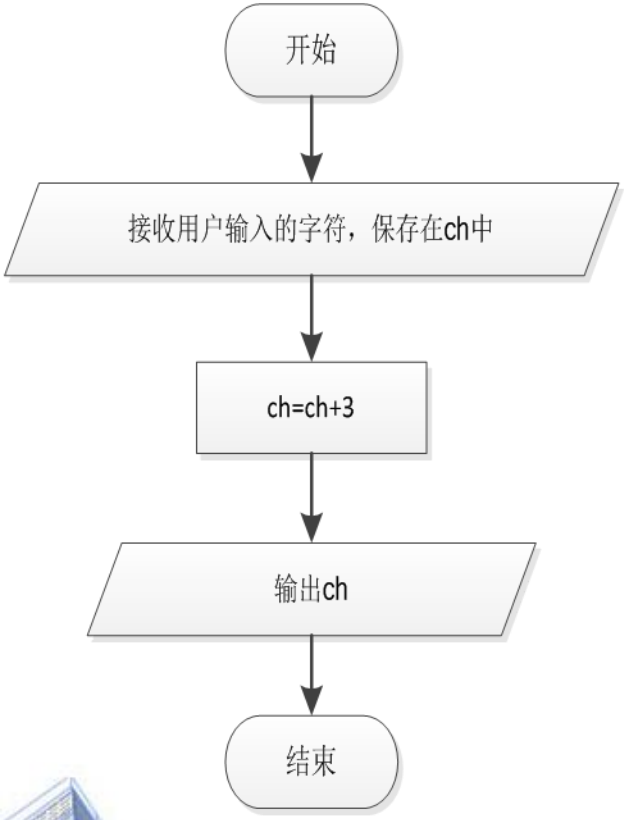


图2-2为字符加密流程图

➢ 4. 程序实现

```
#include <stdio.h>
void main ( )
{
   char ch;    /*定义一个字符变量ch*/
   ch=getchar(); /*调用getchar()函数，从键盘上输
     入一个字符，赋值给ch*/
   ch=ch+3;    /*为字符ch加密 */
   putchar(ch); /*调用putchar()函数，输出加密后
     的ch*/
   putchar('\n');
}
```

单字符输入/输出

- 单字符输入函数getchar( )的一般调用格式为：

  getchar( );

- getchar( )函数为单个字符的输入函数，通常把输入的字符赋值给一个字符变量，构成赋值语句。

  例如：ch=getchar( );

- 单字符输出函数putchar( )的一般调用格式为：

  putchar(ch);

  其中ch是一个字符变量或者常量，也可以是一个ASCII码值（不大于255的整数）。例如：

```
putchar('T'); /*输出一个字符T，是一个字符常量*/
putchar(100); /*输出一个ASCII值是100的字符，即d*/
putchar('\n'); /*输出一个转义字符，即换行符*/
```

输入输出字符

- **scanf( ) 和 printf( )**

  %c

  ```
  char ch;
  scanf("%c", &ch);
  printf("%c", ch);
  ```

- **getchar() 和 putchar()**

  ```
  char ch;
  ch = getchar( );
  putchar(ch);
  ```

  输入输出一个字符

输入输出字符示例

```
# include <stdio.h>
int main(void)
{
    char ch1, ch2;

    ch1=getchar( );
    ch2=getchar( );
    putchar(ch1);
    putchar('#');
    putchar(ch2);
    return 0;
}
```

```
Ab
A#b
```

输入输出字符示例

```
# include <stdio.h>
int main(void)
{
    char ch1, ch2, ch3;
    scanf("%c%c%c", &ch1, &ch2, &ch3);
    printf("%c%c%c%c%c", ch1, '#', ch2, '#', ch3);
    return 0;
}
```

```
AbC
A#b#C
```

```
A bC
A# #b
```

字符运算

- 大小写英文字母转换

'b' - 'a' == 'B' - 'A'

……

'z' - 'a' == 'Z' - 'A'

'm' ←→ 'M'

```
'm'-> 'M'
'M'-> 'm'
```

- 'm' - 'a' + 'A' == 'M'
- 'M' - 'A' + 'a' == 'm'

- 数字字符和数字转换

9 - 0 == '9' - '0'

'9' == 9 + '0'

'8' ←→ 8

```
'8'->8
8->'8'
```

- '8' - '0' == 8
- 8 + '0' == '8'

导例：考核通过了吗？

- **1. 问题描述**

  给出三项考核成绩，判定单项考核和综合考核是否通过。

- **2. 问题分析**

  假如考核通过标准线是60分，将考核成绩与60进行比较，即可判定该项考核是否通过。如果每项考核成绩都大于等于60分，则判定综合考核通过；否则不通过。

- **3. 算法描述**

  接收用户输入的三项考核成绩，存入成绩变量中score1、score2、score3；

  将各项成绩与60进行比较，输出比较结果。

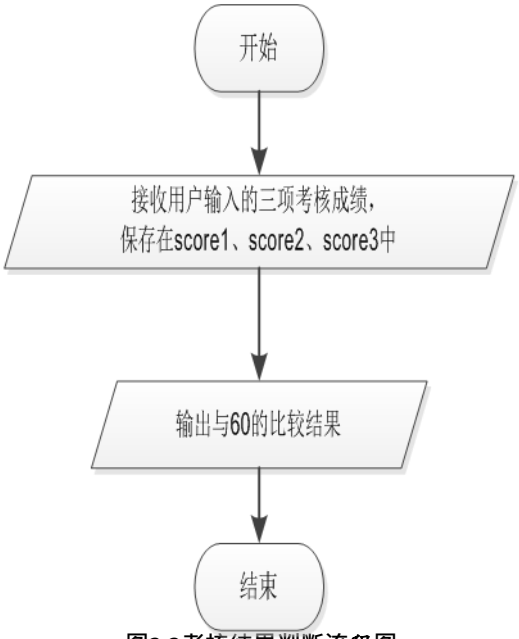
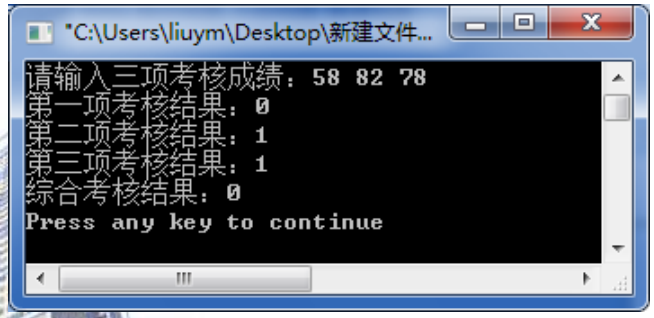


图2-3考核结果判断流程图



➢ 4. 程序实现

```
#include <stdio.h>
 main ()
{
int score1，score2，score3; /*定义各项考核成绩变量*/
/*输入考核成绩*/
  printf("请输入三项考核成绩： ");
  scanf("%d%d%d",&score1,&score2,&score3);
/*输出考核结果*/
  printf("第一项考核结果： %d\n",score1>=60);
  printf("第二项考核结果： %d\n",score2>=60);
  printf("第三项考核结果： %d\n",score3>=60);
  printf("综合考核结果： %d\n",(score1>=60) && (score2>=6
    0) && (score3>=60));
}
```

运算符与表达式

➢ 算术运算符

算术运算符用于各种数值运算，包括5个二元运算符：加（+）、减（-）、乘（*）、除（/）、求余（%），和2个一元运算符：自增（++）、自减（--）。

例如：

34+23、23.4-32、54.3*3、80.1/4、100%34

如果定义如下变量并赋值：

```
int x; float y;
x = 30;y = 53.64;
x+y、x*y、x/23.5
```

注意

/ 整数除整数，得整数

如：1/2 = 0，9/4 = 2

% 针对整型数据

如： 5%6=5，9%4=1，100%4=0

双目运算符两侧操作数的类型要相同

运算符与表达式

- 使用自增（++）、自减（--）这两个一元运算符的运算只需一个操作数，一般用于整型变量的加1、减1操作。
- 例如

```
int x;float y;
x = 30;y = 53.64;
```

x++　x的值加1，变为31。

(x++)+y　x先与y相加，然后x再加1，表达式结果为83.64;

(++x)+y　x先加1，然后再与y相加，结果为84.64。

运算符与表达式

- 关系运算符

关系运算符用于比较运算，包括6个运算符：

大于（>）、小于（<）、大于等于（>=）、

小于等于（<=）、等于（==）和不等于（!=）。

这6个逻辑运算符运算结果为逻辑值：1或0

当条件成立时为真，结果为1；

当条件不成立时为假，结果为0。

区分=和==

运算符与表达式

➢ 逻辑运算符

逻辑运算符用于逻辑运算，包括3个运算符：

非（!）、与（&&）、或（||）

逻辑运算的结果均为逻辑值：1或0。

操作数a	操作数b	a&&b	a||b	!a
0	0	0	0	1
0	非0	0	1	1
非0	0	0	1	0
非0	非0	1	1	0

运算符的优先级

- 在运算表达式中

  算术运算符>关系运算符>逻辑运算符

运算符	功能	优先级
>	大于	高
>=	大于等于	高
<	小于	高
<=	小于等于	高
==	等于	低
!=	不等于	低

运算符	功能	优先级
!	非	高 ↓ 低
&&	与	
||	或	

2.3 C程序的基本控制结构

- 语句是构造C程序的最基本单位，程序运行的过程也就是语句执行的过程，语句执行的次序称为流程控制。
- C程序有三种控制结构，分别是顺序结构、选择分支结构和循环结构。

  顺序结构是指按照语句的书写顺序依次执行；

  选择分支结构是指通过对特定条件的判断，选择执行一个分支；

  循环结构是指在给定条件下，重复执行某段程序，直到条件不满足为止。

导例：计算圆的周长和面积

- **1. 问题描述**

  从键盘输入圆的半径，计算圆的周长和面积。

- **2. 问题分析**

  定义3个float型变量，分别用于存储半径、周长和面积，然后根据圆的周长公式perimeter=2πr，面积公式area= πr^2进行计算。

- **3. 算法描述**

  定义3个float变量r，perimeter和area。

  键盘输入半径r。

  利用公式perimeter=2πr计算周长。

  利用公式area=πr^2计算面积。

  屏幕输出周长perimeter和面积area。

➢ 4. 程序实现

```
#include <stdio.h>
#define  PI 3.14    /*宏定义*/
 main( )
{
   float r,perimeter,area;
   printf("Enter radius:");
   scanf("%f", &r);
   perimeter = 2* PI *r;
   area = PI *r*r;
   printf("Perimeter=%.2f,Area=%.2f\
}
```

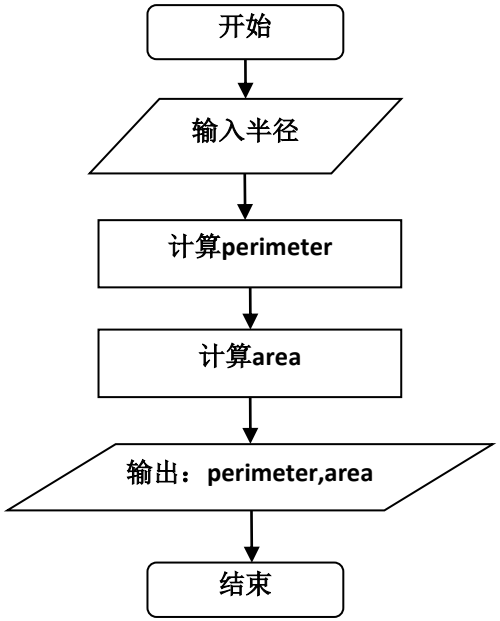
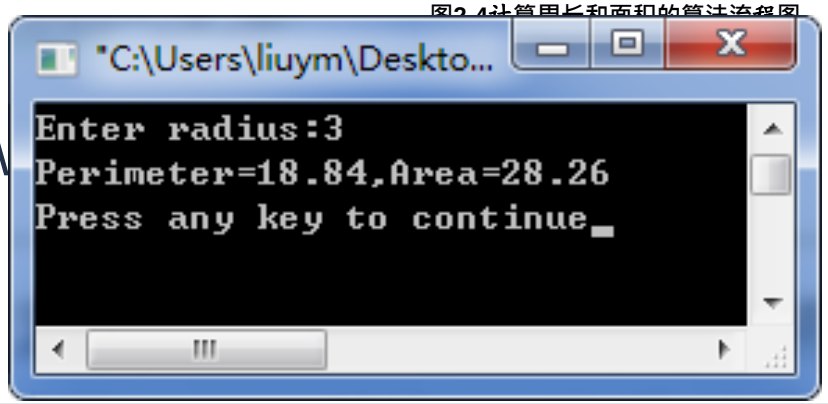


图2.4计算周长和面积的算法流程图



判断数字的奇偶性

- **1、问题描述**

  输入1个整数，判断该数是奇数还是偶数。

- **2、问题分析**

  如果输入的整数能够被2整除，则是偶数，否则是奇数。

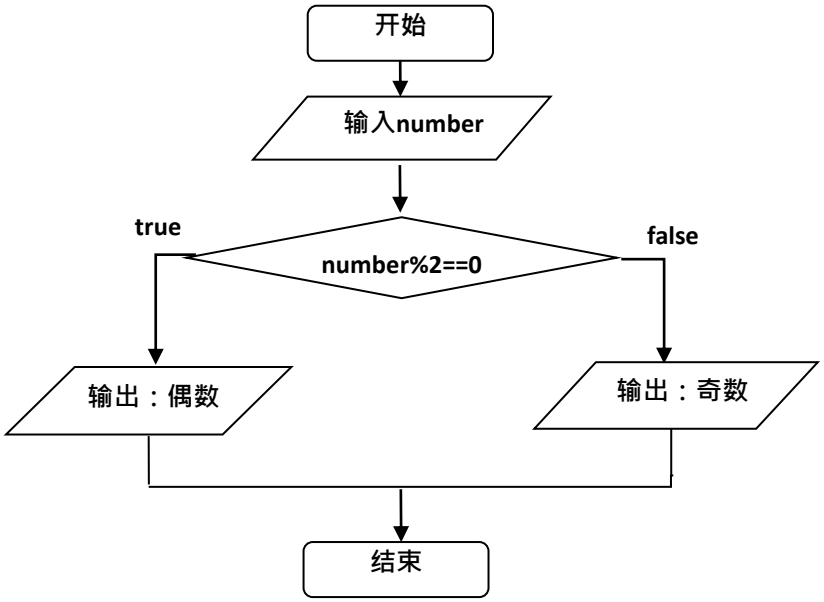
- **3、算法描述**

  输入一个整数保存于整型变量number中。

  判断条件：能被2整除，如果条件成立，则输出：偶数。

  否则，输出：奇数。

```
#include <stdio.h>
int main( )
{   int number;
    printf("Enter a number: ");
    scanf("%d", &number);
    if(number % 2 == 0){
        printf("偶数. \n");
    }
    else{
        printf("奇数. \n");
    }
    return 0;
}
```



此处条件内由于只有一条语句，故{ }可省略

今年是闰年吗？

- **1. 问题描述**

  从键盘输入年份信息，判断是否为闰年。

- **2. 问题分析**

  如果年份能够满足下述任意一个条件，就是闰年；如果两个条件都不满足，就不是闰年。

  条件一：能被4整除且不能被100整除；

  条件二：能被400整除。

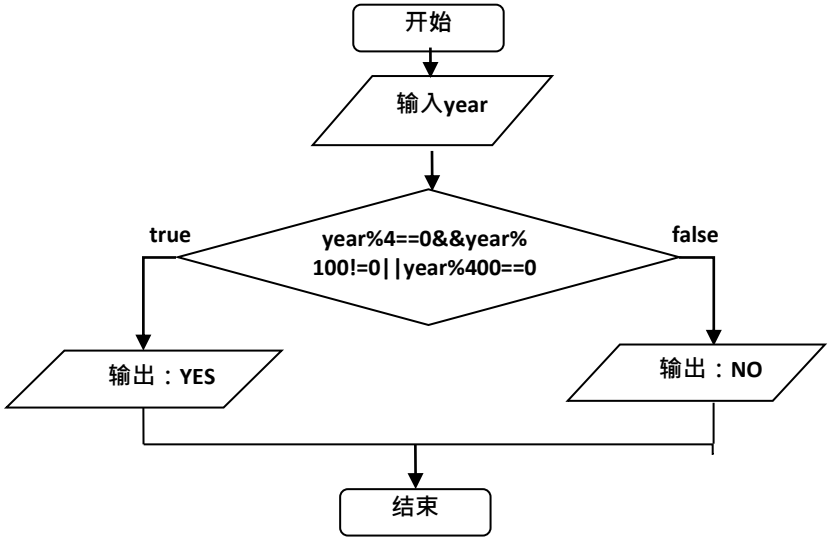
- **3. 算法描述**

  输入年份信息保存于整型变量year中。

  判断条件：能被4整除且不能被100整除或能被400整除，如果条件成立，则输出：是。

  否则，输出：否。

4. 程序实现




```
#include <stdio.h>
 main( )
{
   int year;
   printf("Enter year:");
   scanf("%d ", &year);
   if(year%4==0&&year%100!=0||year%400==0)   /*闰年的条件*/
        printf("YES\n");
   else
        printf("NO\n");
}
```

求解一元二次方程

- **1．问题描述**

  从键盘输入一元二次方程$ax^2+bx+c=0$的3个系数a,b,c，编程计算该方程的解，并输出之。

- **2．问题分析**

  根据判别式b^2-4ac的值不同，一元二次方程的解可分为以下3种情况：

  $b^2-4ac>0$，方程有2个实数解，分别为 $x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$

  $b^2-4ac=0$，方程只有1个实数解，为 $x=-\frac{b}{2a}$

  $b^2-4ac<0$，方程没有实数解

3．算法描述

输入一元二次方程的3个参数a, b, c。

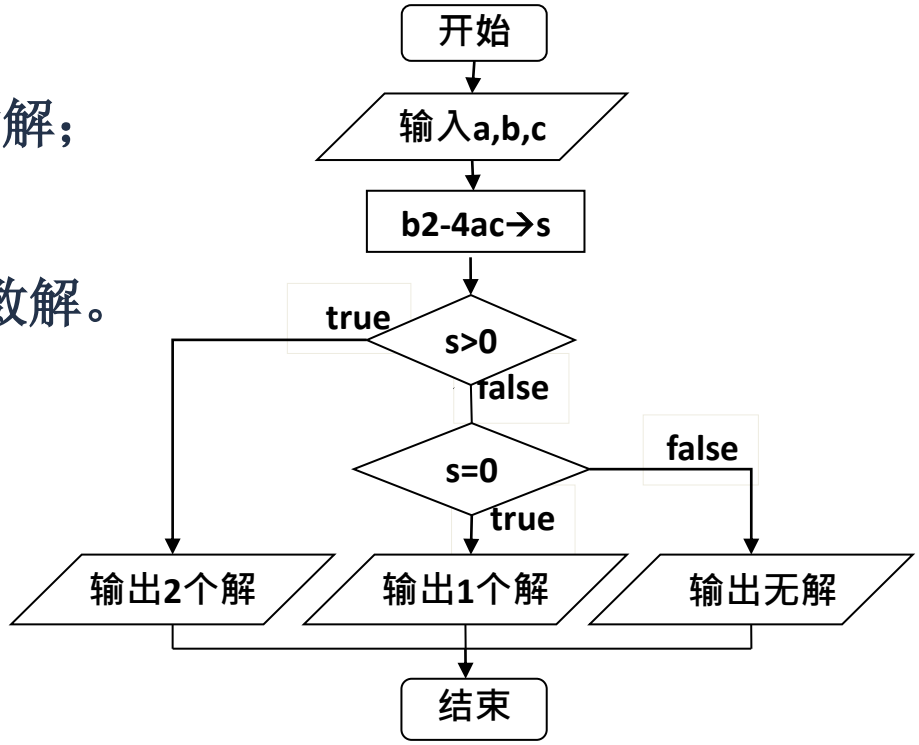
计算判别式b^2-4ac，并将结果存在变量s中。

根据判别式的值进行判断：

如果s>0，输出方程的2个实数解；

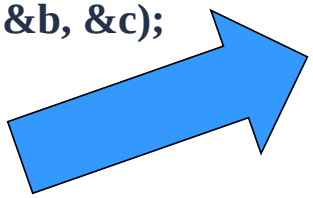
如果s=0，输出方程唯一解；

否则（即s<0），输出没有实数解。



➢ 4、程序实现

```
#include <stdio.h>
#include <math.h>
main()
{
int a, b, c, s;
double x1,x2 ;
printf("Enter 3 integers:");
scanf("%d%d%d", &a, &b, &c);
s=b*b-4*a*c ;
/*条件判断，输出结果*/
}
```



```
/*条件判断，输出结果*/
if (s>0)
{
  x1=(-b+sqrt(s))/(2*a) ;
  x2=(-b-sqrt(s))/(2*a) ;
  printf("x1=%lf, x2=%lf\n", x1,x2) ;
}
else if (s==0)
{
  x1=x2=(-b)/(2.0*a) ;
  printf("x=%lf\n", x1) ;
}
else
  printf("No solution\n");
```

常用数学库函数

- 库函数
  - C语言处理系统提供事先编好的函数，供用户在编程时调用。scanf(), printf(), exp()
  - 在相应的系统文件（头文件）中定义一些必需的信息。
- #include命令
  - 用户调用库函数时，将相应的头文件包含到源程序中。

例如

  - 调用scanf, printf，需要 #include <stdio.h>
  - 调用sqrt，需要 #include <math.h>

常用数学库函数

- 平方根函数 **sqrt(x)**
- 绝对值函数 **fabs(x)**

  fabs(-3.56) 的值为3.56
- 幂函数 **pow(x, n)**：x^n

  pow(1.1, 2) 的值为1.21（即1.1^2）
- 指数函数 **exp(x)**：e^x

  exp(2.3) 的值为$e^{2.3}$
- 以e为底的对数函数 **log(x)**：ln x

  log(123.45) 的值为4.815836
- 以10为底的对数函数 **log10(x)**：$\log_{10}x$

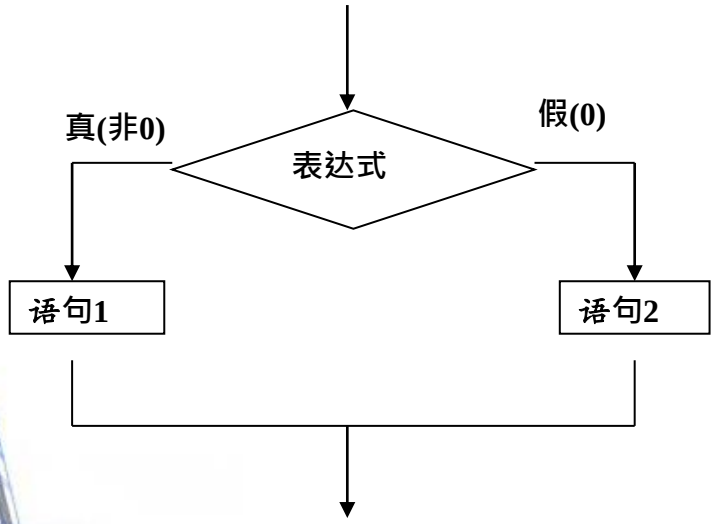
  log10(123.45) 的值为2.091491。

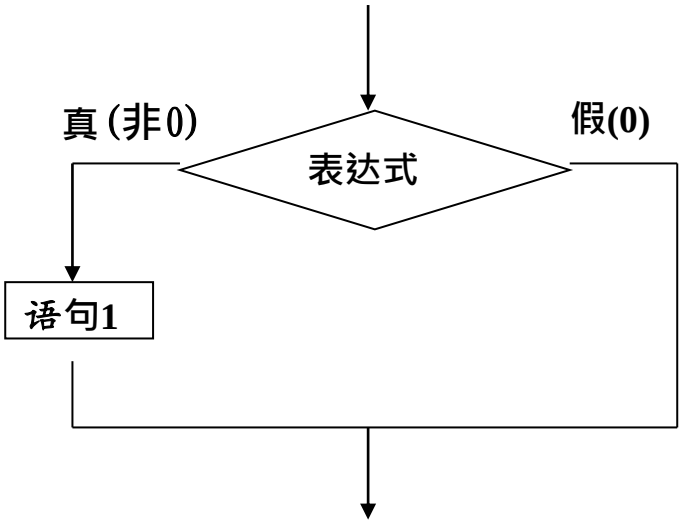
二分支结构和 if-else 语句

```
if (表达式)
    语句1
else
    语句2
```

一条语句

```
if （表达式）
    语句1
```





多分支结构和else – if 语句

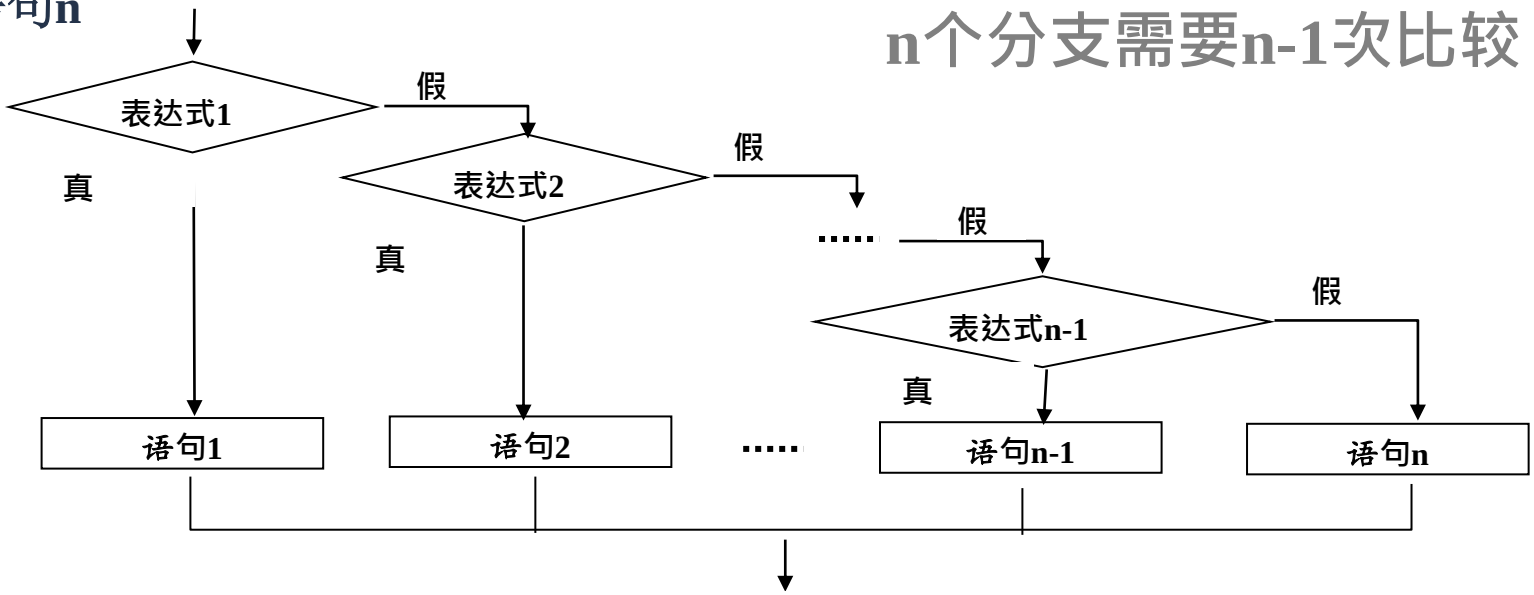
- **else-if 语句是最常用的实现多分支（多路选择）的方法。**

一般格式为：

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

```
if (表达式1)   语句1
else if(表达式2)  语句2
……
else if(表达式n-1)  语句n-1
else 语句n
```

n个分支需要n-1次比较



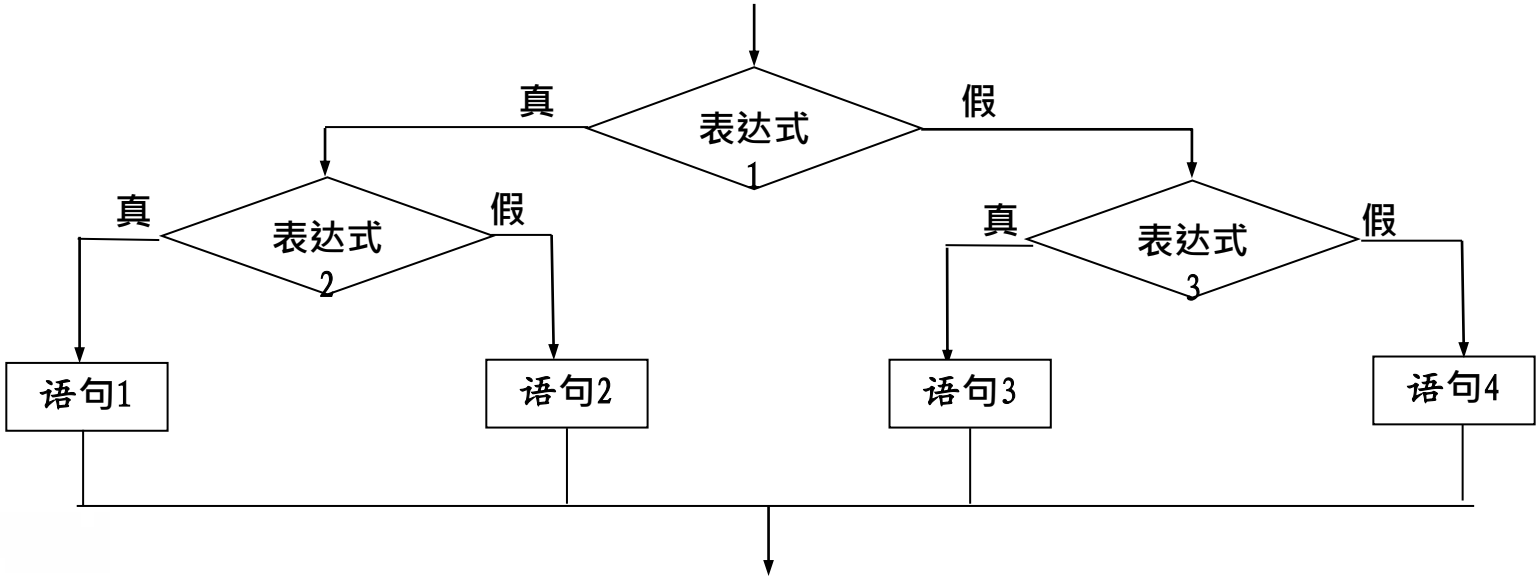
多分支选择结构还可使用switch语句或if-else语句的嵌套来实现。

```
/*使用if-else语句的嵌套来实现求解*/
if (s>0)
{
  x1=(-b+sqrt(s))/(2*a) ;
  x2=(-b-sqrt(s))/(2*a) ;
  printf("x1=%lf, x2=%lf\n", x1,x2) ;
}
else
    if (s==0)
    {
       x1=x2=(-b)/(2.0*a) ;
       printf("x=%lf\n", x1) ;
    }
    else
       printf(“No solution\n”);
```

```
if (表达式)
    语句1      <-- if 语句
else
    语句2      <-- if 语句
```

if-else语句的嵌套的结构

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



四则运算

1、问题描述

求解简单的四则运算表达式。

2、问题分析

输入一个形式如“操作数 运算符 操作数”的四则运算表达式，根据输入的运算符，做相应运算，输出运算结果。

3、算法分析

键盘输入形式如“操作数 运算符 操作数”的四则运算表达式。

根据运算符的不同，做不同的计算。

输出运算结果。

4、程序实现

```
# include <stdio.h>
int main(void) {
    double value1, value2;
    char op;    /*定义字符型的变量*/
    scanf("%lf%c%lf", &value1, &op, &value2);
    if(op == '+')
        printf("=%.2f\n", value1 + value2);
    else if(op == '-')
        printf("=%.2f\n", value1 - value2);
    else if(op == '*')
        printf("=%.2f\n", value1 * value2);
    else if(op == '/')
        printf("=%.2f\n", value1 / value2);
    else
        printf("Unknown operator\n");
    return 0;
}
```

➢ 字符常量

'a'　'z'　'A'　'Z'　'0'　'9'　' '　'\n'

ASCII字符集：列出所有可用的字符

每个字符：惟一的次序值（ ASCII 码）

区分数字 1 和
数字字符 '1'

ASCII码表

符号	10进制	符号	10进制	符号	10进制	符号	10进制
@	64	P	80	`	96	p	112
A	65	Q	81	a	97	q	113
B	66	R	82	b	98	r	114
C	67	S	83	c	99	s	115
D	68	T	84	d	100	t	116
E	69	U	85	e	101	u	117
F	70	V	86	f	102	v	118
G	71	W	87	g	103	w	119
H	72	X	88	h	104	x	120
I	73	Y	89	i	105	y	121
J	74	Z	90	j	106	z	122
K	75	[	91	k	107	{	123
L	76	\	92	l	108	|	124
M	77	]	93	m	109	}	125
N	78	^	94	n	110	~	126
O	79	_	95	o	111	⌂	127

➢ 使用switch语句实现四则运算

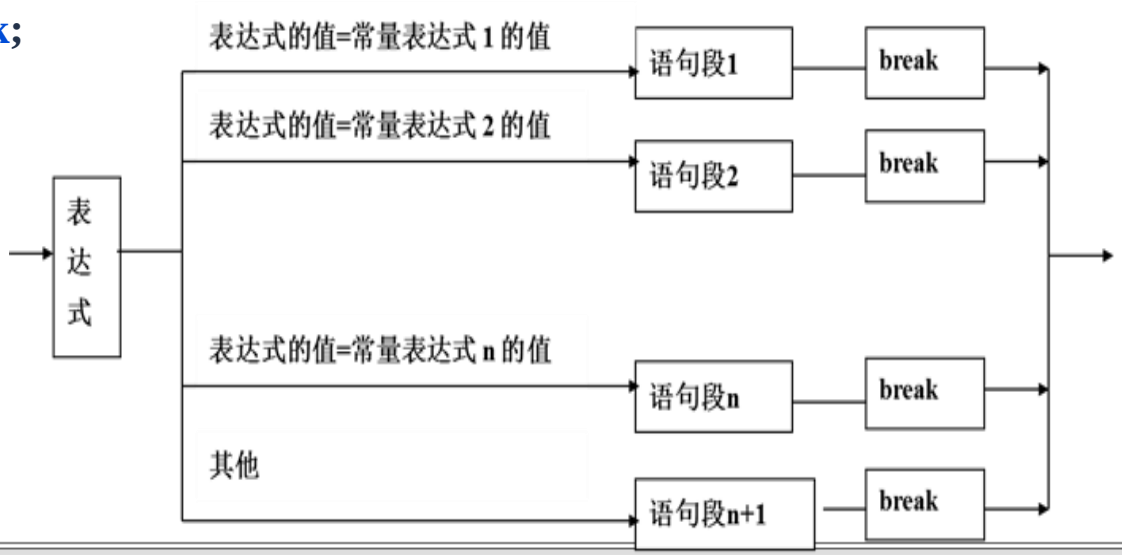
```
# include <stdio.h>
int main(void) {
    double value1, value2;
    char op;
    scanf("%lf%c%lf", &value1, &op, &value2);
     switch (op) {
           case '+': printf("=%.2f\n", value1 + value2); break;
           case '-': printf("=%.2f\n", value1 - value2); break;
           case '*': printf("=%.2f\n", value1 * value2); break;
           case '/': printf("=%.2f\n", value1 / value2); break;
           default:printf("Unknown operator\n"); break;
       }
return 0;
}
```

switch语句

处理多分支选择问题，3种情况

1、在switch语句的每个语句段中都使用break语句

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



计算存款利息

1. 问题描述

银行对整存整取存款期限不同对应的存款利率也不同，键盘输入存款本金和存期，计算到期时的利息及利息与本金的和。

当前整存整取年息利率：

一年：3.25%

二年：3.75%

三年：4.25%

五年：4.75%

2. 问题分析

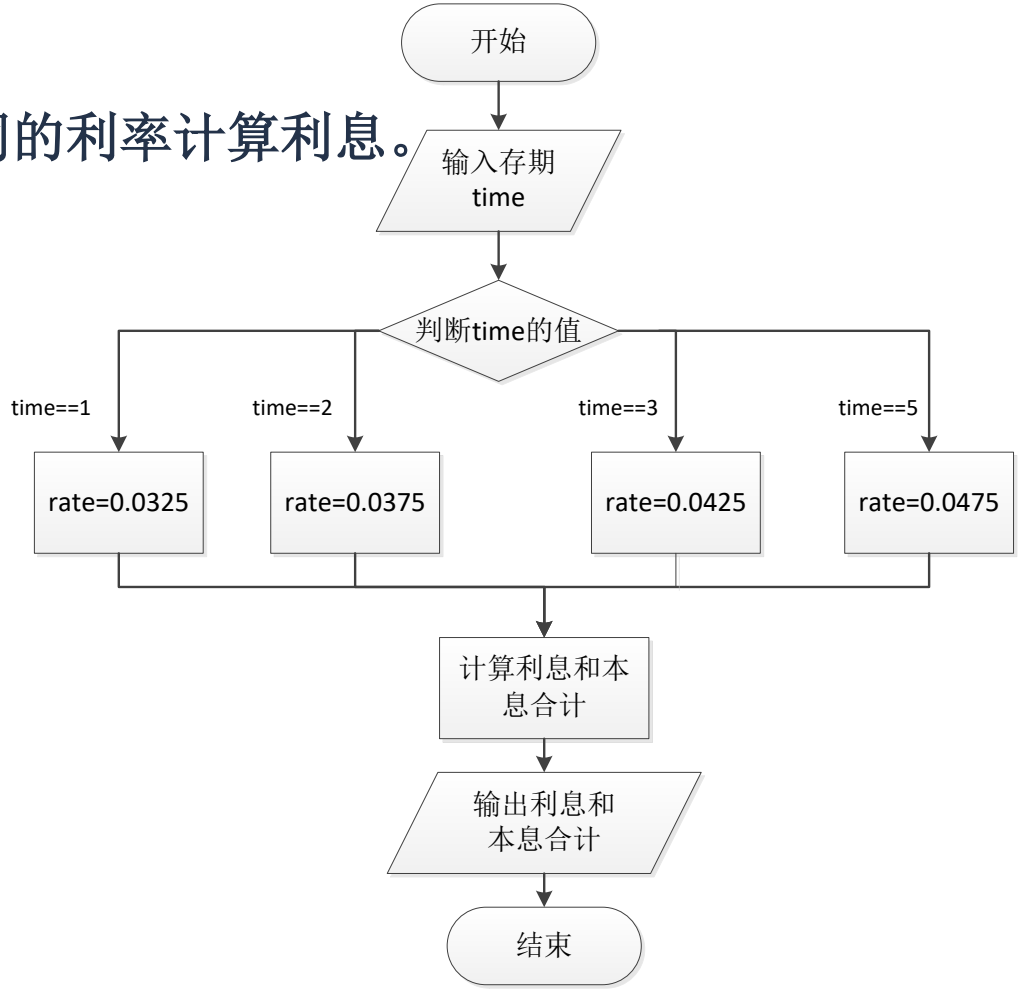
本导例需要根据不同的存款期限决定存款利率，考虑使用switch语句实现多分支选择结构。

3. 算法描述

键盘输入存款本金和存期。

根据存款期限的不同，按照不同的利率计算利息。

输出利息和本息合计。



➢程序实现

```
#include <stdio.h>
main( )
{
  double money, rate, rest, total;
  int time;
  printf("输入本金:");
  scanf("%lf",&money);
  printf("请输入存期:");
  scanf("%d",&time);
/*使用switch语句实现利率的选择*/
 rest=money*rate*time;
 total=money+rest;
 printf("到期利息： %.2f\n",rest);
 printf("本息合计： %.2f\n",total);
}
```

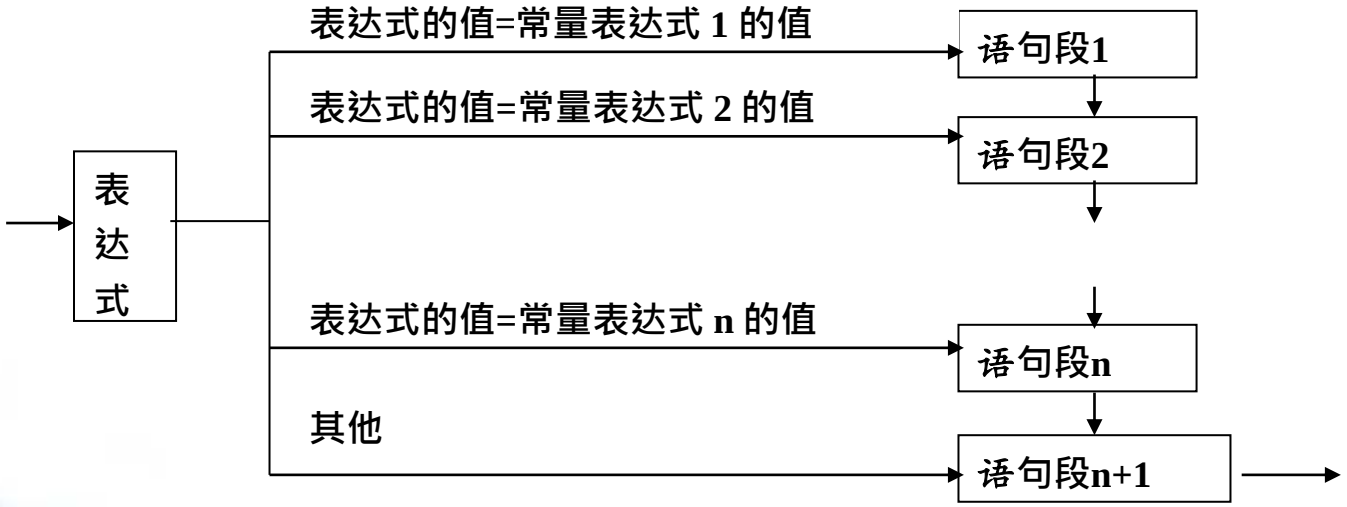
```
switch(time){
    case 1: rate=0.0325; break;
    case 2: rate=0.0375; break;
    case 3: rate=0.0425; break;
    case 5: rate=0.0475; break;
    default: rate=0;
  }
```

➢ 2、在switch中不使用break

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

```
switch(time){
    case 1: rate=0.0325;
    case 2: rate=0.0375;
    case 3: rate=0.0425;
    case 5: rate=0.0475;
    default: rate=0;
  }
```

rate=?



➢ 3、在switch的某些语句段中使用break

输入1个字符，输出该字符所属的种类，如空格或回车、数字字符，其他字符。

```
int main(void)
{  int blank, digit, i, other;  char ch;
   blank = digit = other = 0;
   printf("Enter 10 characters: ");
   ch = getchar();
   switch (ch){
       case ' '  :case '\n':
                 printf("该字符是空格或回车"); break;
       case '0' : case '1' : case '2' : case '3' : case '4' :case '5' :
case '6' : case '7' : case '8' : case '9' :
                printf("该字符是数字字符"); break;
      default:
                printf("该字符是其他字符"); break;
   }
   return 0;
}
```

可以构成三角形吗？

- **1．问题描述**

  输入三条边，判断它们能否构成三角形，若能则指出是何种三角形。

- **2．问题分析**

  给定的三条边，只要两边之和大于第三边即可构成三角形，否则不能构成三角形。在能构成三角形的情况下，如果三边相等，则是等边三角形；如果只有两边相等，则是等腰三角形。可以看出这是多重判断，需要多次运用**if-else**语句来实现。

➢ 3、算法描述

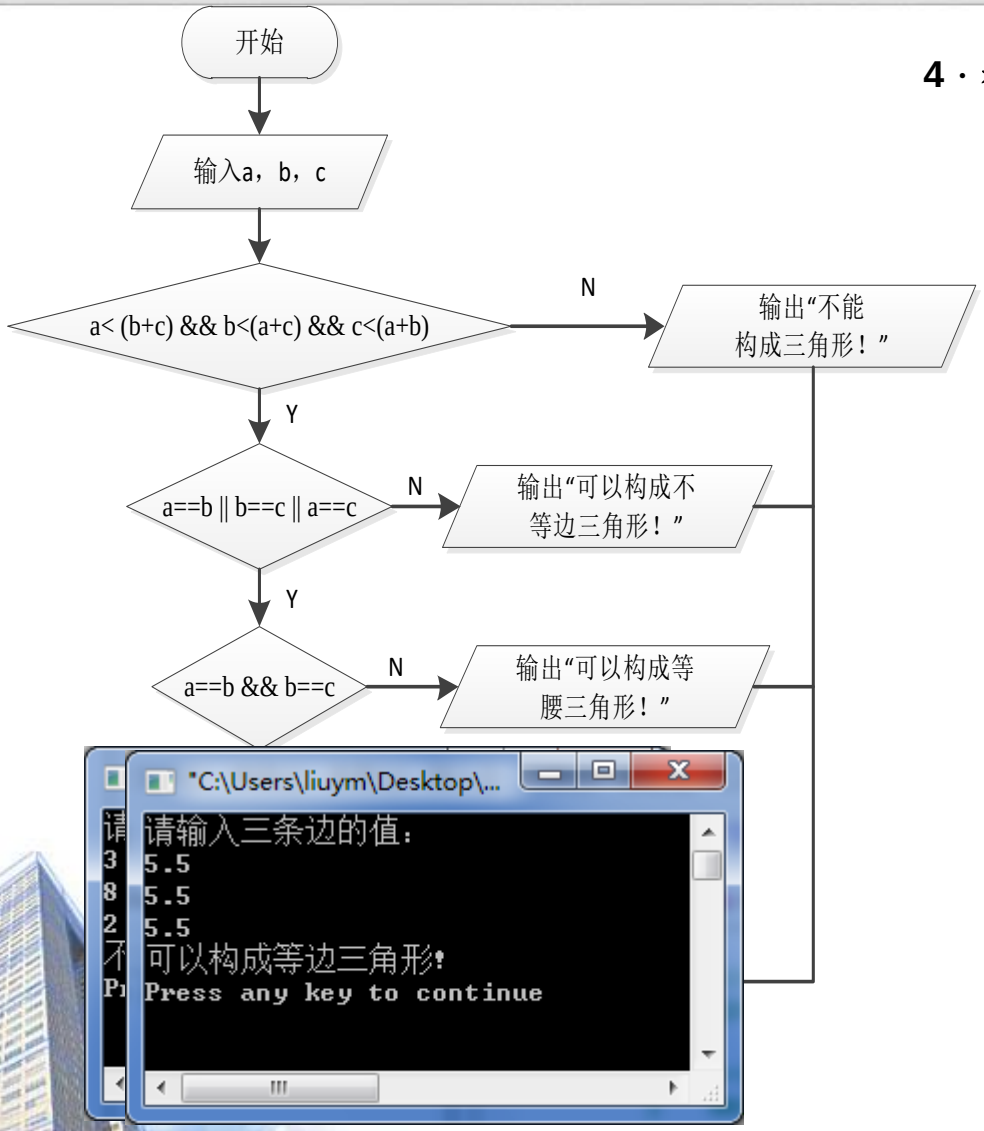
上述问题求解过程以算法的形式描述为：

（1）输入三条边的值，分别保存于变量a、b、c中；

（2）如果a<(b+c)，b<(a+c)，c<(a+b)三个条件同时成立，则执行（3）；否则输出“不能构成三角形！”；

（3）如果a=b，b=c，a=c三个条件中至少有一个成立，则执行（4）；否则输出“可以构成不等边三角形！”；

（4）如果a=b且b=c，则输出“可以构成等边三角形！”；否则输出“可以构成等腰三角形！”；



4．程序实现

```
#include <stdio.h>
main()
{
float a,b,c;
printf("请输入三条边的值：\n");
scanf("%f%f%f",&a,&b,&c);
if(((b+c)<a)||((a+c)<b)||((a+b)<c))
    printf("不能构成三角形!\n");
else
    if(a !=b&&b!=c&&a!=c)
        printf("可以构成不等边三角形!\n");
    else
        if (a==b&&b==c)
            printf("可以构成等边三角形!\n");
        else
            if(a==b||b==c||a==c)
                printf("可以构成等腰三角形!\n");
}
```

练习

1. 输入整数a和b，计算并输出a、b的和与差。
2. 输入华氏温度f，计算并输出相应的摄氏温度c(保留2位小数)。c = 5/9(f-32).
3. 输入四个整数，输出其中的最小值。
4. 输入整数x，若x大于0，y=1；若x等于0，y=0；否则，y=-1，最后输出y。