Fortran95 程序设计

彭国伦 编著

6-0 概 念

- 循环是一种Fortran结构，可允许多次执行一个语句序列。
- 循环结构有两种基本形式：当循环和迭代循环（或称为计数循环）。
- 两种循环的区别：
  - 当循环中的代码循环次数不确定，直到满足某个指定的条件为止。
  - 迭代循环中的循环次数是确定的，重复次数在循环开始前就是已知的。

6-1迭代或计数循环的形式

“循环”：自动连续重复执行某一段程序代码

```
do   counter=Imin,Imax,Istep
…….
…….
…….
End  do
```

最后的数字是计数器的增值，每执行一次循环，counter就会累加上这个数值，可省略，默认为1

计数器的终止数值，counter<=Imax时就会继续重复循环

Counter变量被称为“计数器”，循环重复的次数就根据它的数值而来。变量在刚进入循环时counter被设定为Imin

用来结束循环的程序区块

循环体内的语句每执行一次，控制变量按步长增加一次

- 迭代或计数循环的循环次数：
- 次数=(Imax-Imin+Istep)/Istep

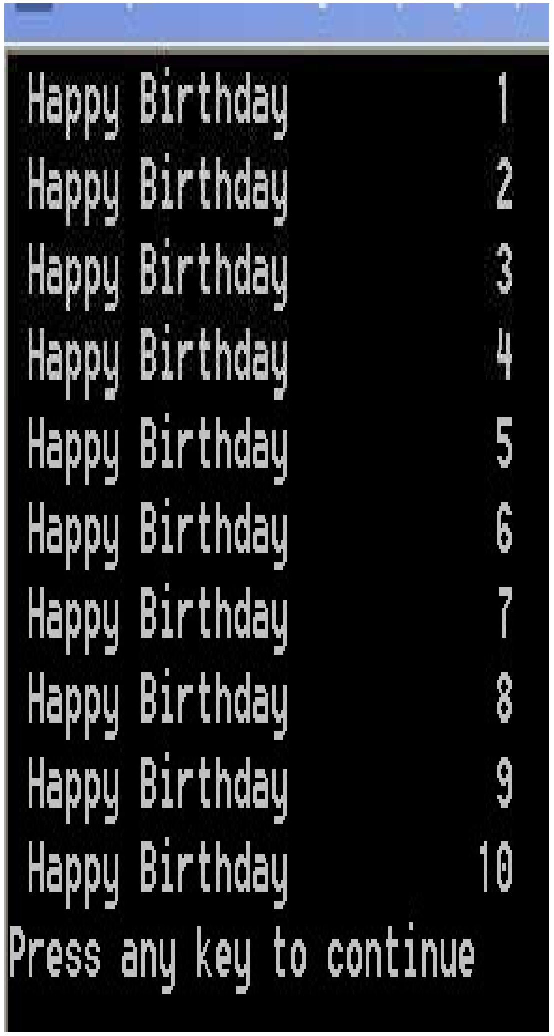
```
DO I=1, 10
     语句1
     ……
     语句n
END DO
```

- 在这个例子中，语句1到语句n将被执行10次。控制变量I第一次为1，第二次为2，以此类推。控制变量在最后一次执行执行语句时为10。在第10次循环后，控制器回到DO语句处，控制变量I增加到11，由于11>10，控制器将转移到END DO语句后面的第一条语句处。

Ex0601.f90

```
program ex0601
implicit none
  integer counter
  integer, parameter :: lines=10
  ! counter<=lines之前会一直重复循环
  ! 每跑一次循环counter会累加1
  do counter=1,lines,1
    write(*,*) "Happy Birthday",counter
  end do

  stop
end
```



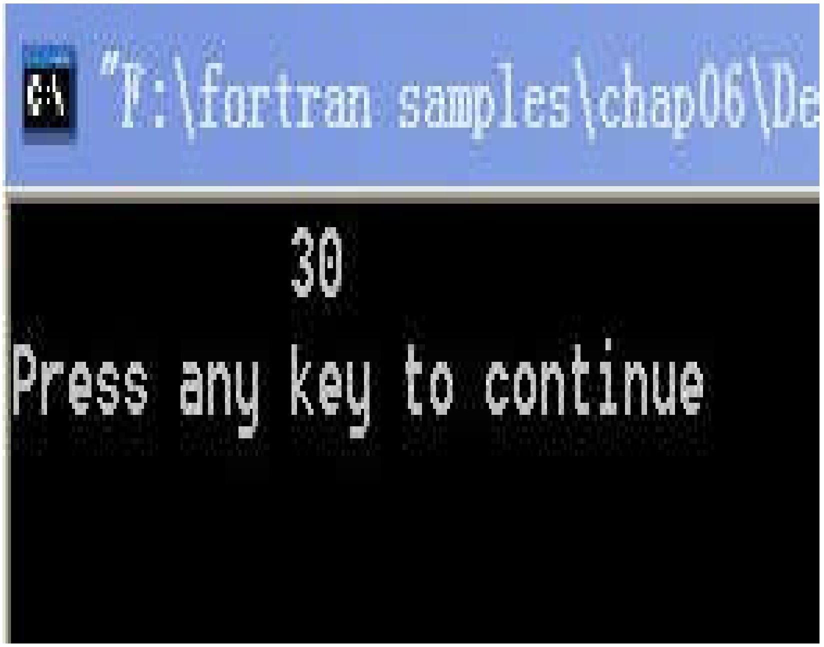
```
 Happy Birthday           1
 Happy Birthday           2
 Happy Birthday           3
 Happy Birthday           4
 Happy Birthday           5
 Happy Birthday           6
 Happy Birthday           7
 Happy Birthday           8
 Happy Birthday           9
 Happy Birthday          10
Press any key to continue
```

Ex0602.f90

```
program ex0602
implicit none
  integer, parameter :: limit=10  ！计数器的上限
  integer counter       ！计数器
  integer :: ans = 0   ！累加器的使用

  do counter=2, limit ,2
    ans = ans + counter
  end do
  write(*,*) ans

  stop
end
```



```
"F:\fortran samples\chap06\De
          30
Press any key to continue
```

6-1迭代或计数循环

- 循环的**增值**可以是**正数**，也可为**负数**，但必须使计数器到达终止值，否则会造成“死循环”。
- 计数器可使用变量，但在循环体中不能改变它的数值，不然在编译时会发生错误。
- DO 和 IF 一样可以是多层嵌套的结构

```
    do  i=1，  10        （第1层循环开始）
        do  i=1，  10        （第2层循环开始）
            do  i=1，  10        （第3层循环开始）
            end  do              （第3层循环结束）
        end  do              （第2层循环结束）
    end  do              （第1层循环结束）
```

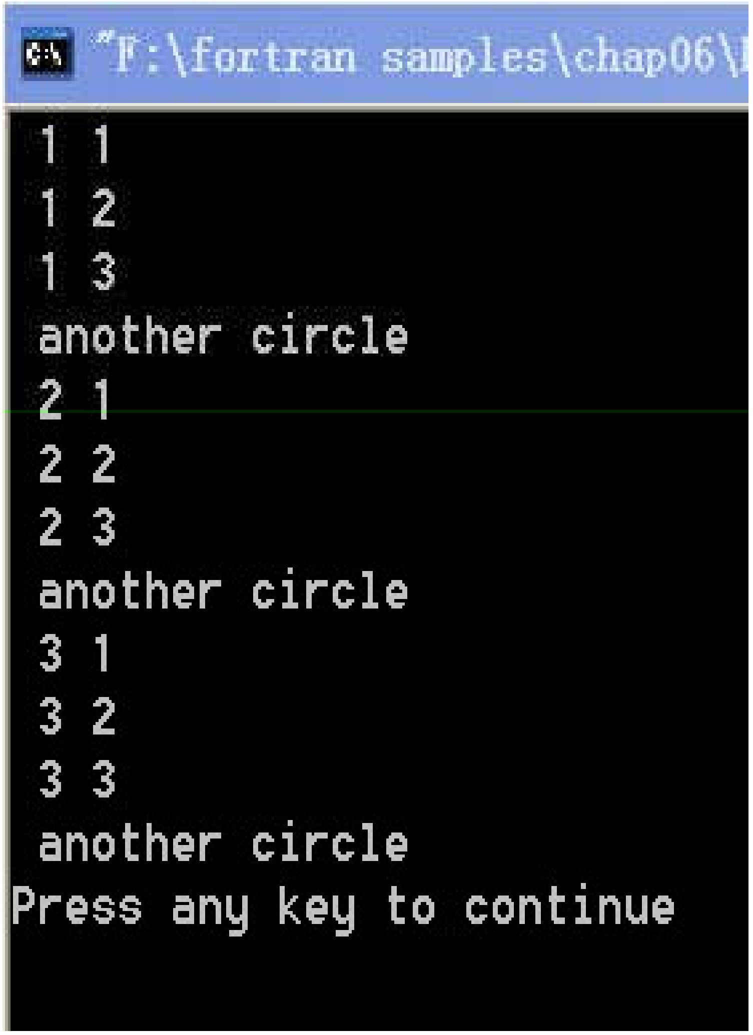
使用嵌套循环要小心，内循环是外循环的好几倍。

Ex0603.f90

```
program ex0603
implicit none
  integer i,j

  do i=1, 3
    do j=1, 3
      write(*, "(I2,I2)") i,j
    end do
    write(*,*) "another circle"
  end do

  stop
end
```



```
"F:\fortran samples\chap06\
 1 1
 1 2
 1 3
 another circle
 2 1
 2 2
 2 3
 another circle
 3 1
 3 2
 3 3
 another circle
Press any key to continue
```

例子1：阶乘函数

- 可以使用DO循环计算阶乘函数，计算正数N的N阶乘的Fortran代码为：

```
N_factorial=1
DO I=1, N
    N_factorial=N_factorial*I
END DO
```

例子2：计算一年中的天数

- 对于平常的年份，从某天开始，一年中的天数为1~365之间的数，而对于闰年来说，是一个1~366之间的数。编写Fortran程序，接收输入的年、月、日的数值，计算1月1日到该日期（包含）的天数。
- 解决方案：
  1. 读取输入的年月日数值。
  2. 检查是不是闰年，必要的话加一天。
  3. 计算天数

PROGRAM doy

```
PROGRAM doy

!  Purpose:
!    This program calculates the day of year corresponding to a
!    specified date.  It illustrates the use of counting loops
!    and the SELECT CASE construct.
!
!  Record of revisions:
!      Date       Programmer          Description of change
!      ====       ==========          =====================
!    11/13/06    S. J. Chapman         Original code
!
IMPLICIT NONE

! Data dictionary: declare variable types, definitions, & units
INTEGER :: day          ! Day (dd)
INTEGER :: day_of_year ! Day of year
INTEGER :: i            ! Index variable
INTEGER :: leap_day     ! Extra day for leap year
INTEGER :: month        ! Month (mm)
INTEGER :: year         ! Year (yyyy)
```

```
! Get day, month, and year to convert
WRITE (*,*) 'This program calculates the day of year given the '
WRITE (*,*) 'current date.  Enter current month (1-12), day(1-31),'
WRITE (*,*) 'and year in that order:  '
READ (*,*) month, day, year

! Check for leap year, and add extra day if necessary
IF ( MOD(year,400) == 0 ) THEN
   leap_day = 1          ! Years divisible by 400 are leap years
ELSE IF ( MOD(year,100) == 0 ) THEN
   leap_day = 0          ! Other centuries are not leap years
ELSE IF ( MOD(year,4) == 0 ) THEN
   leap_day = 1          ! Otherwise every 4th year is a leap year
ELSE
   leap_day = 0          ! Other years are not leap years
END IF
```

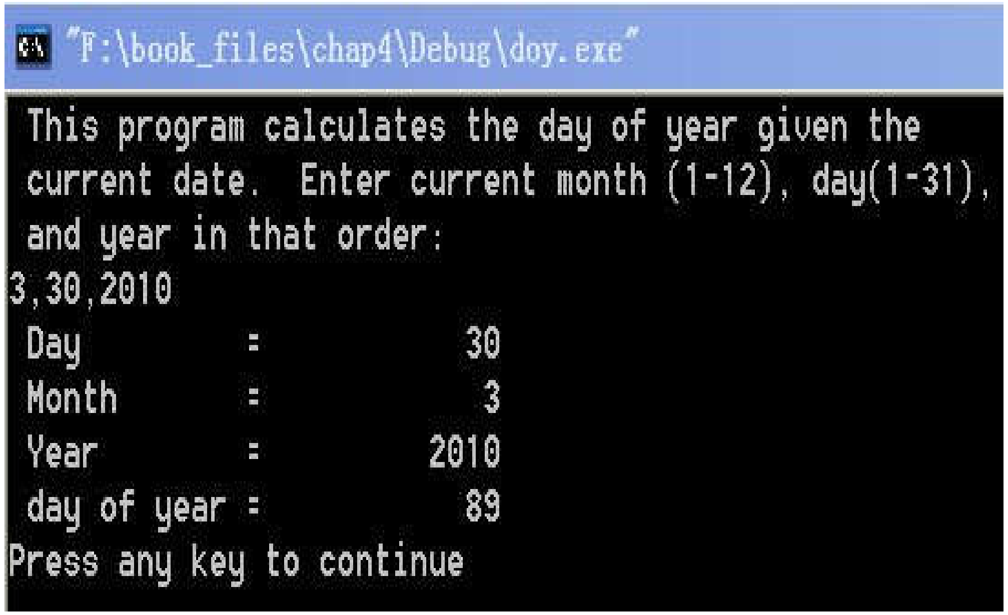
```
! Calculate day of year
day_of_year = day
DO i = 1, month-1

   ! Add days in months from January to last month
   SELECT CASE (i)
   CASE (1,3,5,7,8,10,12)
      day_of_year = day_of_year + 31
   CASE (4,6,9,11)
      day_of_year = day_of_year + 30
   CASE (2)
      day_of_year = day_of_year + 28 + leap_day
   END SELECT

END DO
```

```
! Tell user
WRITE (*,*) 'Day         = ', day
WRITE (*,*) 'Month       = ', month
WRITE (*,*) 'Year        = ', year
WRITE (*,*) 'day of year = ', day_of_year

END PROGRAM doy
```



```
"F:\book_files\chap4\Debug\doy.exe"
 This program calculates the day of year given the
 current date.  Enter current month (1-12), day(1-31),
 and year in that order:
3,30,2010
 Day         =          30
 Month       =           3
 Year        =        2010
 day of year =          89
Press any key to continue
```

例子3：统计分析

- 在科学和工程中通常使用大型的数据系列，每个数据都是对所感兴趣的一些特性的度量值。
- 大多数情况下，对所做出的每一个度量值并不需要仔细查看。
- 当要用一些数值对一组度量值进行合计时，其结果可以告诉我们许多有关综合数据集的信息。
- 此类合计数据有两个，分别是度量值的平均值（算术平均数）和标准方差。

一组数的算术平均数定义为：

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

一组数的标准方差定义为：

$$s = \sqrt{\frac{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}{N(N-1)}}$$

标准方差是对度量值的分散数量的度量，标准方差越大，数据集中的点越分散。

- 实现一个算法，读取一组测量值，计算输入数据组的平均值和标准方差，数据组中的数值可以为正数、负数或零。
- 方案：程序必须能够读入任意个数的测量值，这样就要求用户必须告知输入数值的个数，然后使用DO循环读入这些数值，然后计算这些测量值的平均值和标准方差

PROGRAM stats_1

```
PROGRAM stats_1
!
!  Purpose:
!    To calculate mean and the standard deviation of an input
!    data set, where each input value can be positive, negative,
!    or zero.
!
!  Record of revisions:
!      Date       Programmer          Description of change
!      ====       ==========          =====================
!    11/13/06    S. J. Chapman         Original code
!
IMPLICIT NONE

! Data dictionary: declare variable types, definitions, & units
INTEGER :: i          ! Loop index
INTEGER :: n = 0      ! The number of input samples.
REAL :: std_dev       ! The standard deviation of the input samples.
REAL :: sum_x = 0.    ! The sum of the input values.
REAL :: sum_x2 = 0.   ! The sum of the squares of the input values.
REAL :: x = 0.        ! An input data value.
REAL :: x_bar         ! The average of the input samples.
```

```
! Get the number of points to input.
WRITE (*,*) 'Enter number of points: '
READ  (*,*) n

! Check to see if we have enough input data.
IF ( n < 2 ) THEN ! Insufficient data

   WRITE (*,*) 'At least 2 values must be entered.'

ELSE ! we will have enough data, so let's get it.
```

```
! Loop to read input values.
DO i = 1, n

   ! Read values
   WRITE (*,*) 'Enter number: '
   READ  (*,*) x
   WRITE (*,*) 'The number is ', x

   ! Accumulate sums.
   sum_x  = sum_x + x
   sum_x2 = sum_x2 + x**2

END DO
```

```
! Now calculate statistics.
   x_bar = sum_x / real(n)
   std_dev = SQRT((real(n)*sum_x2 - sum_x**2) /
    (real(n)*real(n-1)))

   ! Tell user.
   WRITE (*,*) 'The mean of this data set is:', x_bar
   WRITE (*,*) 'The standard deviation is:   ', std_dev
   WRITE (*,*) 'The number of data points is:', n

END IF

END PROGRAM stats_1
```

6-2 DO WHILE循环

循环并不一定由计数器的增减来决定是否结束循环，它也可由条件（逻辑运算）来做决定。

```
Do  while （逻辑运算）    ←成立时会一直重复执行循环
   ……
   ……
End  do
```

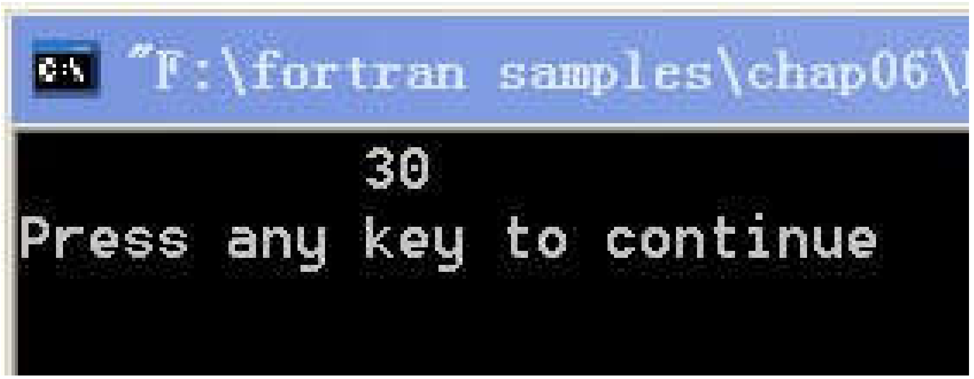
Ex0604.f90

```
program ex0604
implicit none
  integer, parameter :: limit=10  ！计数器的上限
  integer counter       ！计数器
  integer :: ans = 0   ！拿来累加使用

  counter = 2  ！设定计数器初值
  do while( counter <= limit )
    ans = ans + counter
    counter = counter + 2  ！计数器累加
  end do

  write(*,*) ans

  stop
end
```



```
"F:\fortran samples\chap06\
           30
Press any key to continue
```

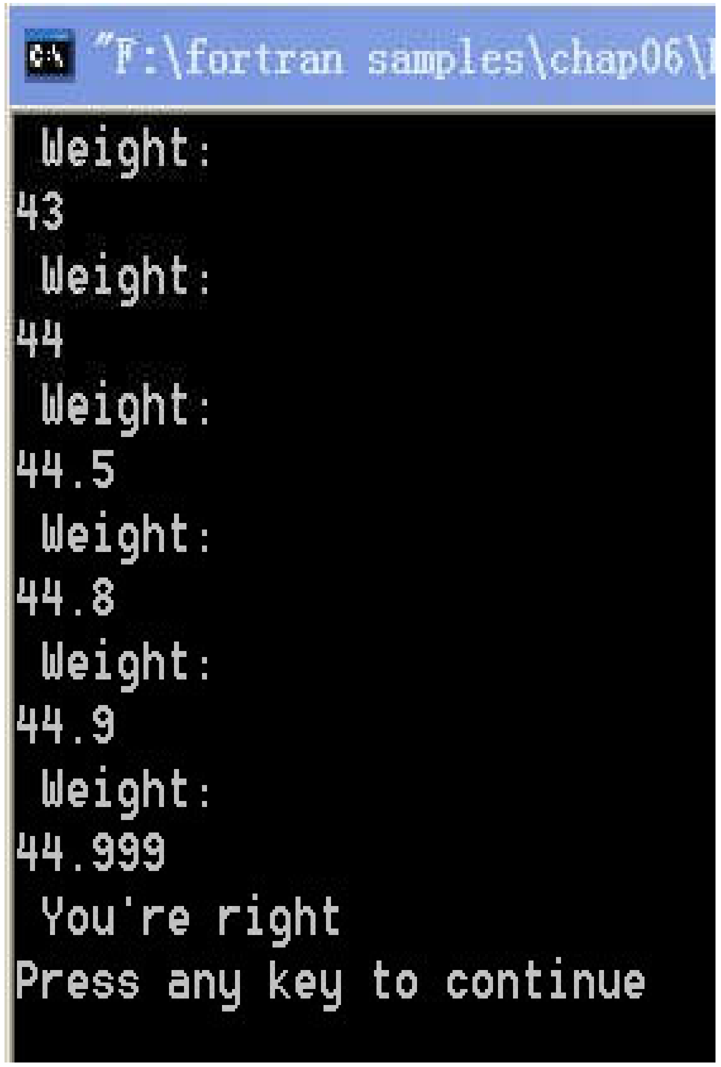
Ex0605.f90

```
program ex0605
implicit none
  real, parameter :: weight=45.0
  real, parameter :: e = 0.001
  real :: guess = 0.0

  do while( abs(guess-weight) > e )
    write(*,*) "Weight:"
    read(*,*) guess
  end do

  write(*,*) "You're right"

  stop
end
```



"F:\fortran samples\chap06\

```
 Weight:
43
 Weight:
44
 Weight:
44.5
 Weight:
44.8
 Weight:
44.9
 Weight:
44.999
 You're right
Press any key to continue
```

6-3 当循环

- 当循环是当满足某些条件时，语句块不确定地重复执行。一般形式为：

```
DO
      ……
      IF (逻辑表达式) EXIT
      ……
END DO
```

- 在好的结构化程序中，当循环应该有单一的入口点和出口点。入口点就是DO语句，出口点就是EXIT语句。当循环中只有一个出口点可以帮助确认当循环在所有情况下都是正确的。

例子4：统计分析

- 目的与例子3相同：读入一组数据，然后计算其平均值和标准方差。
- 要求使用IF <条件> EXIT语句
- 方案：必须能够读入任意多个数据，也必须用某种方法告诉程序没有要输入的数据了。
- 假设：所有的输入值都是正数或零，使用负的输入数值作为没有要读取的数据的标记。

```
PROGRAM stats_2
!
!  Purpose:
!    To calculate mean and the standard deviation of an input
!    data set containing an arbitrary number of input values.
!
!  Record of revisions:
!      Date       Programmer          Description of change
!      ====       ==========          =====================
!    11/10/06    S. J. Chapman        Original code
!
IMPLICIT NONE

! Data dictionary: declare variable types, definitions, & units
INTEGER :: n = 0      ! The number of input samples.
REAL :: std_dev = 0. ! The standard deviation of the input samples.
REAL :: sum_x = 0.   ! The sum of the input values.
REAL :: sum_x2 = 0.  ! The sum of the squares of the input values.
REAL :: x = 0.       ! An input data value.
REAL :: x_bar        ! The average of the input samples.
```

```
! While Loop to read input values.
DO
   ! Read in next value
   WRITE (*,*) 'Enter number: '
   READ  (*,*) x
   WRITE (*,*) 'The number is ', x

   ! Test for loop exit
   IF ( x < 0 ) EXIT

   ! Otherwise, accumulate sums.
   n      = n + 1
   sum_x  = sum_x + x
   sum_x2 = sum_x2 + x**2
END DO
```

```
! Calculate the mean and standard deviation
x_bar = sum_x / real(n)
std_dev = SQRT( (real(n) * sum_x2 - sum_x**2) /
    (real(n) * real(n-1)) )

! Tell user.
WRITE (*,*) 'The mean of this data set is:', x_bar
WRITE (*,*) 'The standard deviation is:   ', std_dev
WRITE (*,*) 'The number of data points is:', n

END PROGRAM stats_2
```

增加一个意外判断

```
!  Check to see if we have enough input data.
IF ( n < 2 ) THEN ! Insufficient information

   WRITE (*,*) 'At least 2 values must be entered!'

ELSE ! There is enough information, so
     ! calculate the mean and standard deviation

   x_bar = sum_x / real(n)
   std_dev = SQRT( (real(n) * sum_x2 - sum_x**2) / (real(n)*real(n-1)))

   ! Tell user.
   WRITE (*,*) 'The mean of this data set is:', x_bar
   WRITE (*,*) 'The standard deviation is:   ', std_dev
   WRITE (*,*) 'The number of data points is:', n

END IF

END PROGRAM stats_2
```

6-4 循环的流程控制

有两个可用于控制当循环和计数DO循环操作的附加语句

6-3-1 CYCLE

CYCLE：略过当前循环，直接进入下一次的循环

6-3-2 EXIT

EXIT：直接强制跳出当前运行的循环

6-3-3 署名的循环

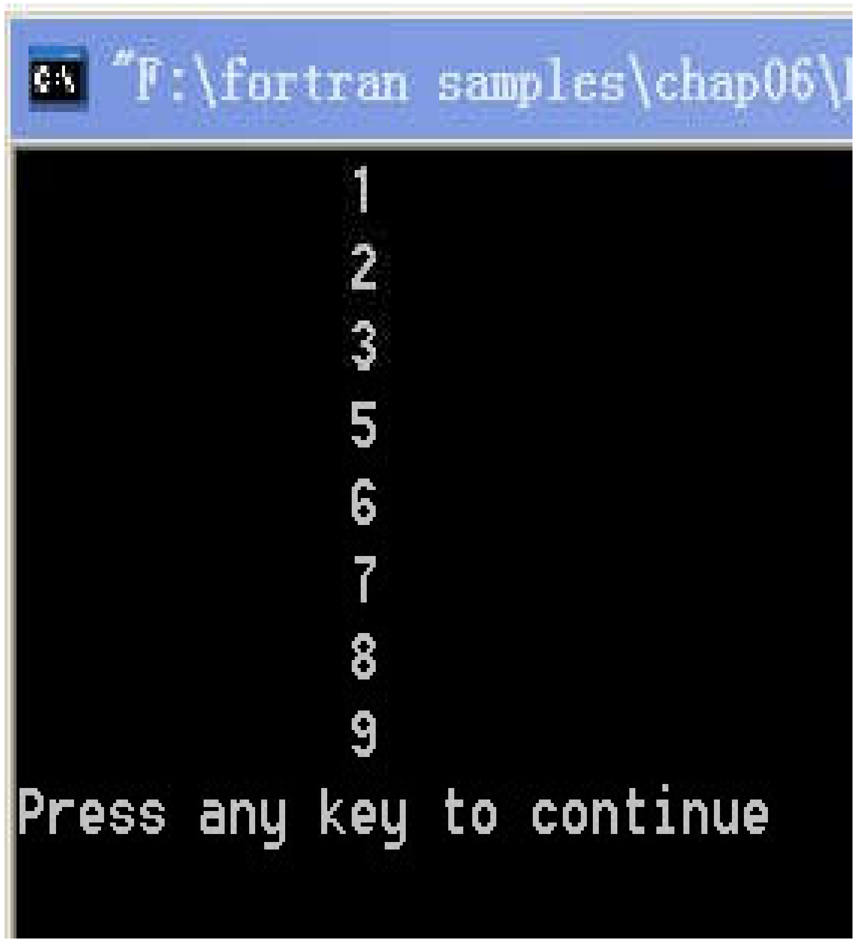
在多层循环中不易出错

program ex0606

```
program ex0606
implicit none
  integer :: dest = 9
  integer floor

  do floor=1, dest
    if ( floor==4 ) cycle
    write(*,*) floor
  end do

  stop
end
```



"F:\fortran samples\chap06\

```
           1
           2
           3
           5
           6
           7
           8
           9
Press any key to continue
```

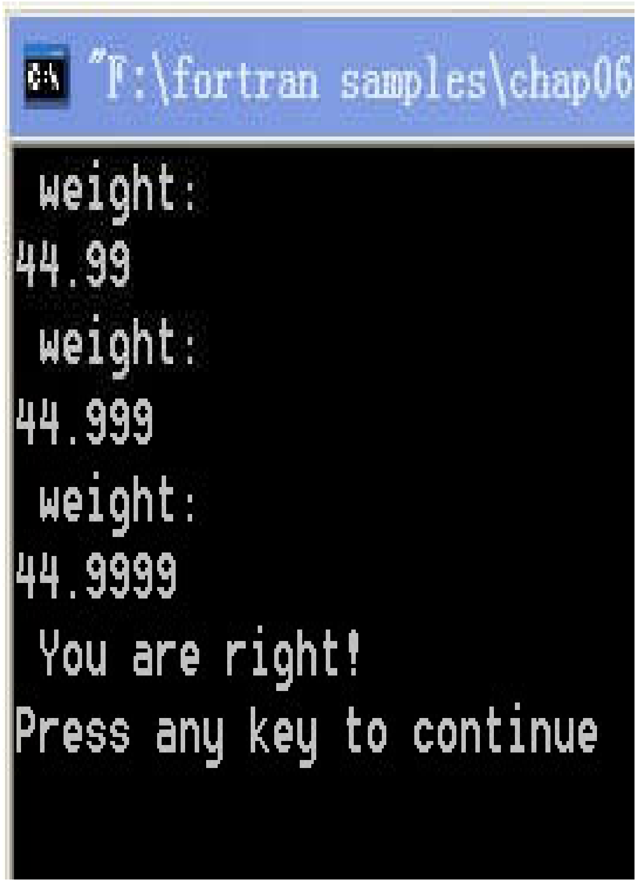
program ex0607

```
program  ex0607
implicit none
  real, parameter :: weight=45.0
  real, parameter :: error=0.0001
  real :: guess = 0.0

  do while( .true. )
    write(*,*) "weight:"
    read(*,*) guess
    if ( abs(guess-weight)<error ) exit
  end do

  write(*,*) "You are right!"

  stop
end
```



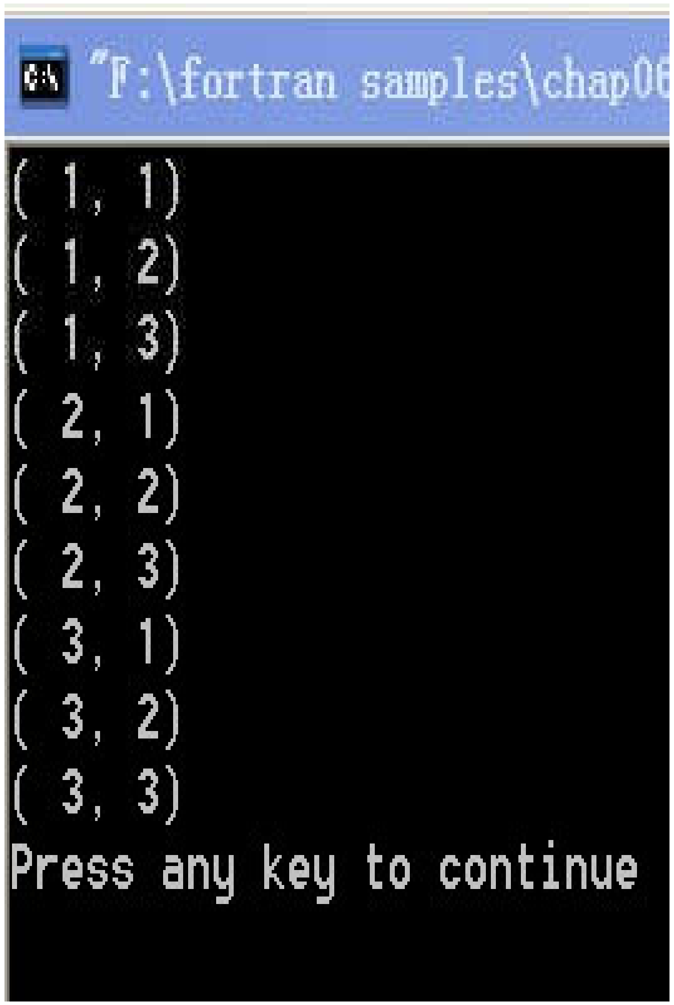
```
"F:\fortran samples\chap06
 weight:
44.99
 weight:
44.999
 weight:
44.9999
 You are right!
Press any key to continue
```

program ex0608

```
program  ex0608
implicit none
  integer  ::  i,j

  outter: do i=1,3
    inner:  do j=1,3
      write(*, "('(',i2,',',i2,')')" ) i,j
    end do inner
  end do outter

  stop
end
```

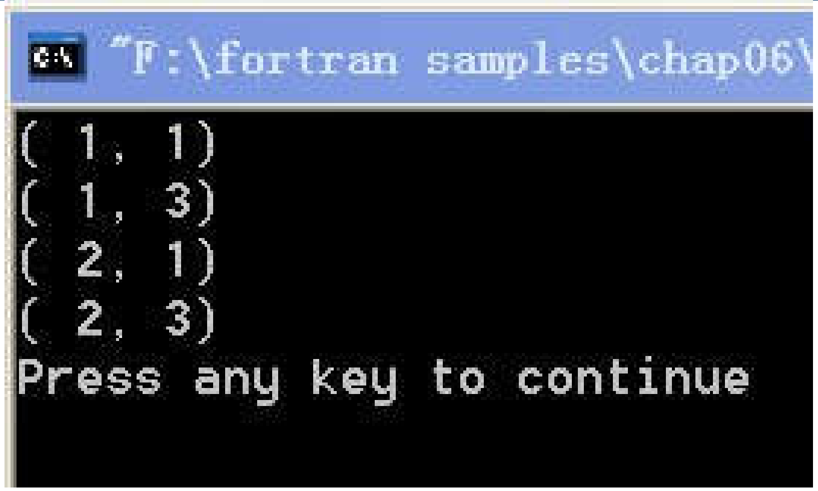

"F:\fortran samples\chap06

```
( 1, 1)
( 1, 2)
( 1, 3)
( 2, 1)
( 2, 2)
( 2, 3)
( 3, 1)
( 3, 2)
( 3, 3)
Press any key to continue
```

program ex0609

```
program  ex0609
implicit none
  integer  ::  i,j

  loop1: do i=1,3
    loop2: do j=1,3
      if ( i==3 ) exit  loop1  ！跳离loop1循环
      if ( j==2 ) cycle loop2  ！重做loop2循环
      write(*, "('(',i2,',',i2,')')" ) i, j
    end do loop2
  end do loop1
  stop
end
```



```
( 1, 1)
( 1, 3)
( 2, 1)
( 2, 3)
Press any key to continue
```

求等差数列1+2+3+4+…+99+100的值

```
program  ex0610
implicit none
  integer counter
  integer :: ans = 0

  do counter = 1, 100
    ans = ans + counter
  end do

  write(*,*) ans

  stop
end
```

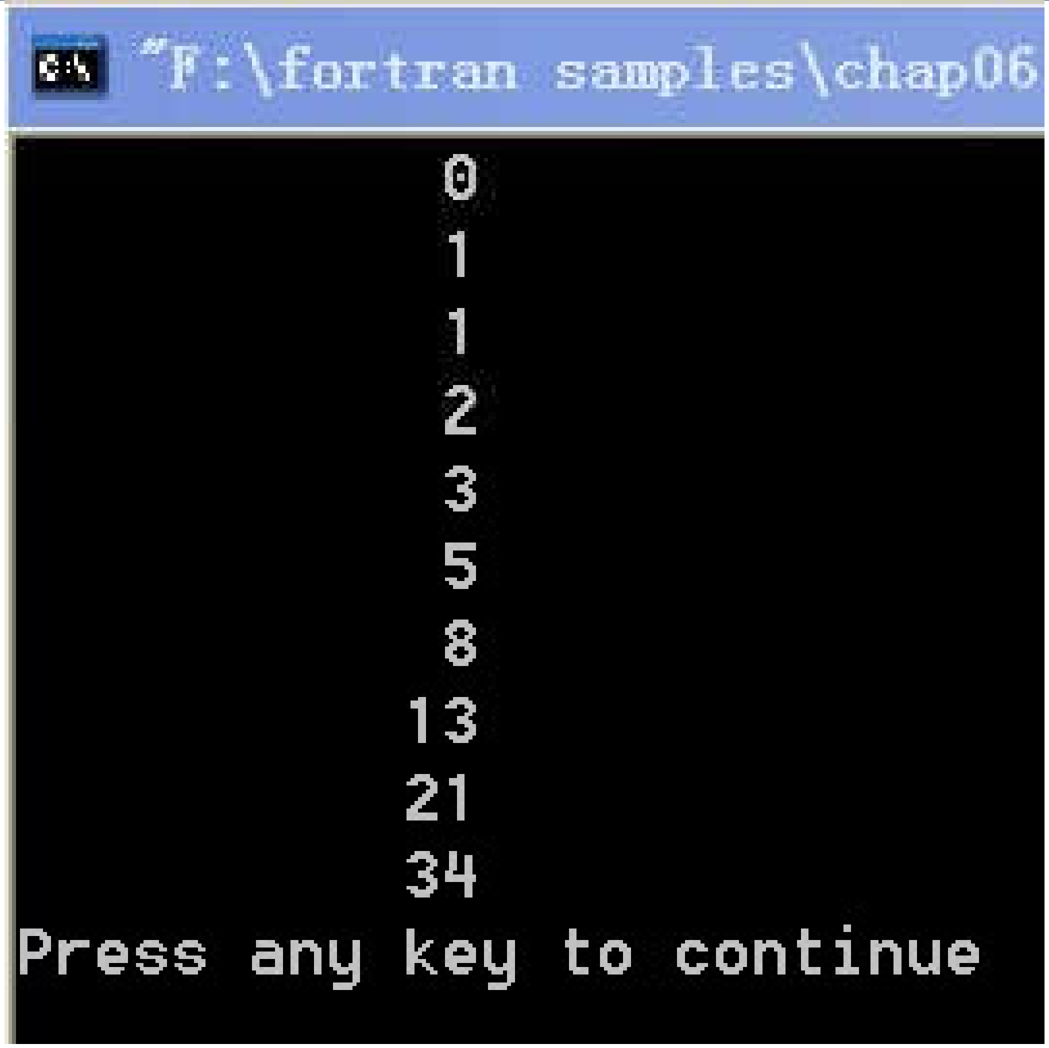
费氏数列 $f_0=0$，$f_1=1$　$n>1$　$f_n=f_{n-1}+f_{n-2}$

```
program  ex0611
implicit none
  integer counter
  integer :: fn_1 = 1
  integer :: fn_2 = 0
  integer :: fn = 0

  write(*,*) fn_2
  write(*,*) fn_1

  do counter = 3, 10
    fn = fn_2 + fn_1
      write(*,*) fn
      fn_2 = fn_1
      fn_1 = fn
  end do

  stop
end
```



```
           0
           1
           1
           2
           3
           5
           8
          13
          21
          34
Press any key to continue
```

6-5 字符内置字符函数

1. IACHAR(C)：接收一单个的字符c，返回其在ASCII字符集中相应位置的整数。
2. ACHAR(ival)：返回ASCII码中相应的字符。
3. LEN(STR1)：字符串的长度
4. LEN_TRIM(STR1)：不包含尾部空格的长度
5. TRIM(STR1)：返回被截去尾部空格的字符串

简单的密码加密

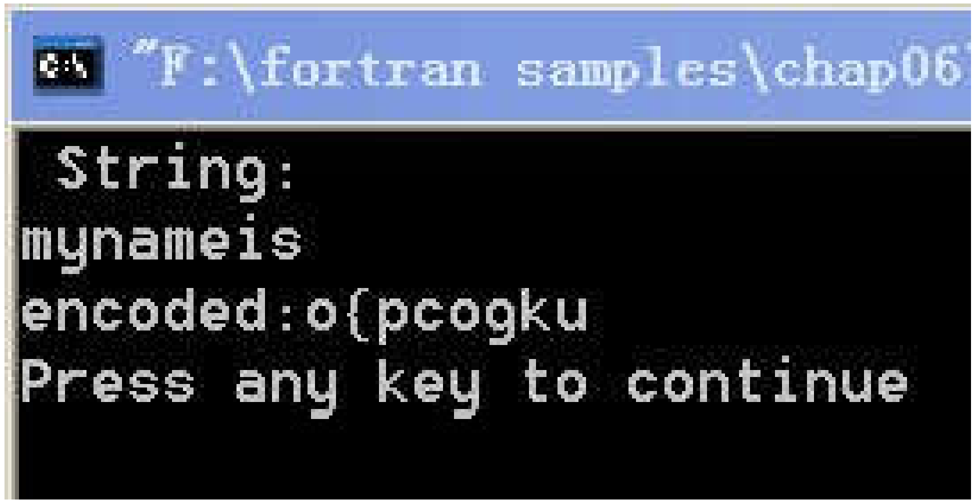
```
program ex0612
implicit none
  integer i
  integer strlen
  integer, parameter :: key = 2
  character(len=20) :: string

  write(*,*) "String:"
  read(*,*) string
  strlen = len_trim(string)

  do i = 1, strlen
    string(i:i) = char( ichar(string(i:i)) + key )
  end do

  write(*,"('encoded:',A20)") string

  stop
end
```



```
 String:
mynameis
encoded:o{pcogku
Press any key to continue
```

简单的密码解密

```
program ex0613
implicit none
  integer i
  integer strlen
  integer, parameter :: key = 2
  character(len=20) :: string

  write(*,*) "Encoded string:"
  read(*,*) string
  strlen = len_trim(string)

  do i = 1, strlen
    string(i:i) = char( ichar(string(i:i)) - key )
  end do

  write(*,"('String:',A20)") string

  stop
end
```

小型的计算器程序

```
program ex0614
implicit none
  real a,b,ans
  character :: key = 'y' ! 为了至少进入循环1次

  do while( key=='y' .or. key=='Y' )
    read(*,*) a
     read(*,"(A1)") key
     read(*,*) b
```

```
select case(key)
case('+')
  ans = a+b
case('-')
  ans = a-b
case('*')
  ans = a*b
case('/')
  ans = a/b
case default
  write(*,"('Unknown operator ',A1)") key
  stop
end select
```

```
 write(*,"(F6.2,A1,F6.2,'=',F6.2)") a,key,b,ans
    write(*,*) "(Y/y) to do again. (Other) to exit."
    read(*,"(A1)") key
  end do
  stop
end
```

例子5：字符串的比对

```
PROGRAM compare
!
!  Purpose:
!    To compare two strings to see if they are equivalent,
!    ignoring case.
!
!  Record of revisions:
!      Date       Programmer          Description of change
!      ====       ==========          =====================
!    11/14/06    S. J. Chapman        Original code
!
IMPLICIT NONE

! Data dictionary: declare variable types, definitions, & units
INTEGER :: i                  ! Loop index
CHARACTER(len=20) :: str1    ! First string to compare
CHARACTER(len=20) :: str1a   ! Copy of first string to compare
CHARACTER(len=20) :: str2    ! Second string to compare
CHARACTER(len=20) :: str2a   ! Copy of second string to compare
```

```
! Prompt for the strings
WRITE (*,*) 'Enter first string to compare:'
READ (*,*) str1
WRITE (*,*) 'Enter second string to compare:'
READ (*,*) str2

! Make copies so that the original strings
! are not modified
str1a = str1
str2a = str2
```

```
! Now shift lower case letters to upper case.
DO i = 1, LEN(str1a)
   IF ( str1a(i:i) >= 'a' .AND. str1a(i:i) <= 'z' ) THEN
      str1a(i:i) = ACHAR ( IACHAR ( str1a(i:i) ) - 32 )
   END IF
END DO
DO i = 1, LEN(str2a)
   IF ( str2a(i:i) >= 'a' .AND. str2a(i:i) <= 'z' ) THEN
      str2a(i:i) = ACHAR ( IACHAR ( str2a(i:i) ) - 32 )
   END IF
END DO
```

```
! Compare strings and write result
IF ( str1a == str2a ) THEN
   WRITE (*,*) "'", str1, "' = '", str2, "' ignoring case."
ELSE
   WRITE (*,*) "'", str1, "' /= '", str2, "' ignoring case."
END IF

END PROGRAM compare
```

例子6：小球射程

- 假定小球的初始速度为20m/s，从初始位置(0,0)处，初始角度为θ被抛出。设计、编写并测试程序，确定小球从抛出时开始到再次到达地面为止行进的水平距离。并针对不同的初始角度，确定出最大射程。

$$y(t) = y_0 + v_{y0}t + \frac{1}{2}gt^2 \qquad v_{x0} = v_0\cos\theta$$

$$x(t) = x_0 + v_{x0}t \qquad v_{y0} = v_0\sin\theta$$

$$0 = 0 + v_{y0}t + \frac{1}{2}gt^2$$

$$0 = \left(v_{y0} + \frac{1}{2}gt\right)t$$

$$t_2 = -\frac{2v_{y0}}{g}$$

$$Range = x(t_2) = x_0 + v_{x0}t_2$$

$$= -\frac{2v_{x0}v_{y0}}{g} = -\frac{2v_0^2}{g}\cos\theta\sin\theta$$

```
PROGRAM ball
!
!  Purpose:
!    To calculate distance traveled by a ball thrown at a specified
!    angle THETA and at a specified velocity VO from a point on the
!    surface of the earth, ignoring the effects of air friction and
!    the earth's curvature.
!
!  Record of revisions:
!      Date       Programmer          Description of change
!      ====       ==========          =====================
!    11/14/06    S. J. Chapman         Original code
!
IMPLICIT NONE

! Data dictionary: declare constants
REAL, PARAMETER :: DEGREES_2_RAD = 0.01745329 ! Deg ==> rad conv.
REAL, PARAMETER :: GRAVITY = -9.81            ! Accel. due to gravity (m/s)
```

```
! Data dictionary: declare variable types, definitions, & units
INTEGER :: max_degrees  ! angle at which the max rng occurs
    (degrees)
REAL :: max_range       ! Maximum range for the ball at vel v0 (meters)
REAL :: range           ! Range of the ball at a particular angle (meters)
REAL :: radian          ! Angle at which the ball was thrown (in radians)
INTEGER :: theta        ! Angle at which the ball was thrown (in degrees)
REAL :: v0              ! Velocity of the ball (in m/s)

! Initialize variables.
max_range = 0.
max_degrees = 0
v0 = 20.
```

```
loop: DO theta = 0, 90

   ! Get angle in radians
   radian = real(theta) * DEGREES_2_RAD

   ! Calculate range in meters.
   range = (-2. * v0**2 / GRAVITY) * SIN(radian) * COS(radian)

   ! Write out the range for this angle.
     WRITE (*,*) 'Theta = ', theta, ' degrees; Range = ', range,  ' meters'

   ! Compare the range to the previous maximum range.  If this
   ! range is larger, save it and the angle at which it occurred.
   IF ( range > max_range ) THEN
      max_range = range
      max_degrees = theta
   END IF

END DO loop
```

```
! Skip a line, and then write out the maximum
! range and the angle at which it occurred.
WRITE (*,*) ' '
WRITE (*,*) 'Max range = ', max_range, ' at ', &
   max_degrees, ' degrees'

END PROGRAM ball
```

6-6 循环的调试

- 计数DO循环中的大多数错误时由于循环参数的错误。可以在适当的地方添加WRITE语句，发现循环中的错误。
- 当循环中的错误通常与用于控制其作用的逻辑表达式的错误有关。可以使用WRITE语句检查当循环中的IF (逻辑表达式) EXIT语句。

作业

P98

1、2、3、4

5. 修改例子6的程序，读入某一特定位置的重力加速度，计算小球在该加速度下的最大射程。分别以9.8、9.7、9.6的重力加速度运行程序，重力吸引的减小对小球的射程有什么影响？对小球抛出的最佳角度有什么影响？再分别以10、20、30的初始速度运行程序，看看初始速度对最佳角度有什么影响。

6. 编写程序，提示用户输入0~1023之间的十进制整数，并将数据转换为等价的二进制数。程序应该给用户显示出相应的二进制数。采用十进制的256、63、140和768对程序进行测试。

作业

7. 假设一个生物学家进行一项实验，她测量在不同的培养基下某一特定类的细菌无性繁殖的速率。实验显示在培养基A中，细菌每90min繁殖一次，培养基B中细菌每120h繁殖一次。假定实验开始时在每种培养基中都放置单个的细菌。编写程序，计数并输出从实验开始到24h后每隔6h，在各个培养基中出现的细菌个数。
8. RMS平均值：均方根（RMS）平均值是另外一种计算一组数的平均数的方法，定义为这组数平方的算术平均值的平方根。编写程序，接收任意多数目的正数输入值，计算这组数的RMS平均值。
9. 描述理想气体的参数有：绝对压力P、体积V和绝对温度T，三者间的关系为：PV=nRT。假定理想气体的抽样在273K的温度下包含1mol的分子，编程回答下列问题：
   - 输出气体压力从1~1001kPa，间隔为100kPa的气体体积。
   - 假设气体温度增加到300K，气体体积如何随着同样范围内压力的改变而改变。

10. 轨道计算。人造卫星绕着地球而运行，人造卫星的轨道形成一个椭圆形，地球位于该椭圆的其中一个焦点上。人造卫星的轨道可以极坐标的形式表示为

$$r = \frac{p}{1 + e\cos\theta}$$

其中：

r和θ为人造卫星相对地球中心的距离和角度；p为确定轨道大小的参数；e为表示轨道离心率的参数。

假定人造卫星的量值参数p=1200km。编写一个程序，如果人造卫星的离心率分别为(a) e= 0；(b) e=0.25；(c) e=0.5时，计算作为θ的函数的人造卫星相对于地球中心的距离，其中r和e均为输入值。