FORTRAN 90

Lecturer : Rafel Hekmat Hameed
University of Babylon
Subject : Fortran 90
College of Engineering
Year : Second B.Sc.
Mechanical Engineering Dep.

Subprograms

Fortran 90 has two types of subprograms, functions and subroutines.

Functions

Functions are used by referring to their name (exactly as you would refer to the intrinsic functions sin(x)) and they produce one answer. A fortran function, or function subprogram, has the following syntax

```
type FUNCTION function-name (arg1, arg2, ..., argn)
IMPLICIT NONE
[specification part]
[execution part]
[subprogram part]
END FUNCTION function-name
```

type is a Fortran 90 type (*e.g.*, INTEGER,REAL, LOGICAL, etc) , function name is a Fortran 90 identifier , arg1, …, arg n are *formal arguments*.

Somewhere in a function there has to be one or more assignment statements like this:
function-name = expression
where the result of expression is saved to the name of the function.

Example

Write a program to read n integers number and then use an integer function to calculate average of n numbers.

```
PROGRAM NUMBERS
IMPLICIT NONE
INTEGER::N,X,SUM
READ(*,*) N
DO I=1,N
READ(*,*) X
SUM=SUM+X
ENDDO
WRITE(*,50) AVERAGE(SUM,N)
50 FORMAT(2X,'AVERAGE=',1X,i5)
END
INTEGER FUNCTION AVERAGE(SUM,N)
IMPLICIT NONE
INTEGER::N,SUM
AVERAGE=SUM/N
END FUNCTION AVERAGE
```

Example

Write a program to read n integers number and then use an integer function to calculate the factorial of n.

```
program fact
implicit none
integer:: factorial,n
read*, n
write(*,7) factorial(n)
7 format (2x,i5)     ;        end
function factorial(n)
implicit none
integer::fa,j,n,factorial
fa=1
```

```
do j=1,n
fa=fa*j        ;        enddo
factorial=fa          ;        end function factorial
```

Example

Write a fortran 90 function subprogram, which transform the Cartesian coordinate presentation (x,y) to the polar coordinate (r,θ), if you know

$$\sin\theta = y/r \ , \ \cos\theta = x/r \ , \ r = (x^2+y^2)^{1/2}$$

Note: θ must be in degree

```
implicit none
real::x,y,f,r,f1,theta
read(*,*)x,y
r=f(x,y)
theta=f1(r,x)
write(*,50)r, theta
50 format(2x,"r=",1x,f9.5,2x,"theta=",1x,f6.3)
end

real function f(x,y)
real::x,y
f=sqrt(x**2+y**2)
end

real function f1(r,x)
real,parameter::pi=3.14159
real::r,x,ff
ff=acos(x/r)
f1=(180*ff)/pi
end
```

Example

Write a function subprogram to evaluate the maximum number of matrix A with rank 2 and shape (/4,4/).

$$A=\begin{bmatrix} 5 & 3 & 7 & 8 \\ 7 & 6 & 2 & 14 \\ 9 & 4 & 8 & 7 \\ 1 & 12 & 1 & 2 \end{bmatrix}$$

```
implicit none
integer,parameter::n=4
integer,dimension(n,n)::a
integer::fmax
data a/5,7,9,1,3,6,4,12,7,2,8,1,8,14,7,2/
write(*,6)"max no.=",fmax(a)
6 format(2x,a,1x,i4)
end

integer function fmax(a)
integer,parameter::n=4
integer,dimension(n,n)::a
integer::i,j,max
max=a(1,1)
do i=1,n
do j=1,n
if(a(i,j).gt.max)then
max=a(i,j)
endif
enddo ; enddo
fmax=max
end
```

Example

Write a fortran 90 function subprogram to check if the matrix A is singular or non-singular. Use select case statement.

$$A=\begin{bmatrix}7 & 9 & 5\\ 5 & 6 & 12\\ 3 & 1 & 4\end{bmatrix}$$

```
Program determinant
implicit none
Integer, parameter::n=3, m=5
Integer, dimension (n,m):: a
Integer:: i, j, det1, det2, det          ;        character(len=30)::ff
Read(*,*) ((a(i,j),j=1,n),i=1,n)
Do i=1,n        ;        do j=1,n-1
a(i,n+j)=a(I,j)  ;        enddo ;        enddo
do i=1,n        ;det1=1        ;        det2=1
do j=1,n
det1=det1*a(j,i+j-1)
det2=det2*a(j,2*n-j-i+1)       ;enddo
det=det+det1-det2    ;        enddo
write(*,6) ff(det)
6 format (3x,a)        ;        end

function ff(det)
character(len=30)::ff
integer::det
select case (det==0)
case(.true.)
ff="singular"
case (.false.)
ff="non-singular"
end select     ;        end
```