

INFORMATICA

Antes:

- Resolución del examen:
- Clase de hoy:
 - Funciones intrínsecas
- **¡Aviso!**
 - Si estáis perdidos, el momento de ponerse al día es ahora. A partir de la clase de la semana que viene todo se complica.
 - Próxima clase es el 8 de Noviembre. Tenéis casi 15 días.

Funciones intrínsecas.

- Son funciones predefinidas que se pueden utilizar sin declarar previamente.
- Las funciones pueden tener argumentos de tipo:
 - Numérico:
 - Conversión de tipo
 - Truncamiento
 - Redondeo
 - Matemáticas
 - Array
 - Matemáticas
 - Búsqueda
 - Dimensiones
 - Character

Funciones intrínsecas.

- Funciones numéricas. Conversión de tipo
 - **Real**(x[,k]) \longrightarrow convierte x a `real(kind=k)`
 - Tipo del argumento (x): `integer`, `real` o `complex`
 - Tipo de la función: `real`
 - **dbble**(x) \longrightarrow convierte x a `real*8`
 - Tipo del argumento (x): `integer`, `real` o `complex`
 - Tipo de la función: `real*8`

(Los corchetes significan opcional)

Funciones intrínsecas.

- Funciones numéricas. Conversión de tipo

```
program prueba_intrinsecas
```

```
integer :: a  
real(8) :: xd  
real :: x
```

```
a = 2
```

```
x = real(a)
```

```
xd = real(a,KIND=8)
```

```
write(*,*) "a", a  
write(*,*) "x", x  
write(*,*) "xd", xd
```

```
end program
```

```
a      2  
x      2.00000000  
xd     2.000000000000000000
```

Funciones intrínsecas.

- Funciones numéricas. Conversión de tipo

- **aimag**(z)

Parte imaginaria del número complejo z

- Tipo del argumento: `complex`
- Tipo de la función: `real`

- **cmplx**(x[,y][,k])

Número complejo de parte real x e imaginaria y y kind=k

- Tipo del argumento: `real` o `complex`
- Tipo de la función: `complex`

Funciones intrínsecas.

- Funciones numéricas. Conversión de tipo

`real` `:: x` \rightarrow `cmplx(x)` = $x + i0.0$

`real` `:: x, y` \rightarrow `cmplx(x, y)` = $x + iy$

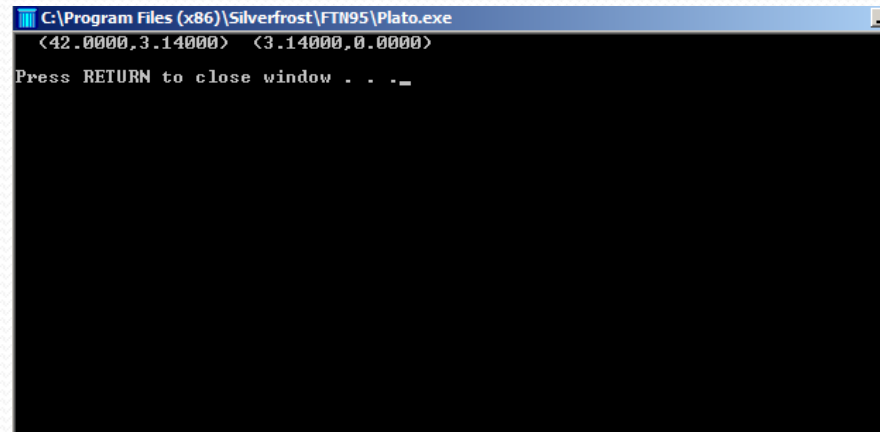
`complex` `:: z` \rightarrow `cmplx(z)` = $Re(z) + iIm(z)$

```
program test_cmplx

  integer :: i = 42
  real    :: x = 3.14
  complex :: z

  z = cmplx(i, x)
  write(*,*) z, cmplx(x)

end program test_cmplx
```



```
C:\Program Files (x86)\Silverfrost\FTN95\Plato.exe
<42.0000,3.14000> <3.14000,0.00000>
Press RETURN to close window . . . _
```

Funciones intrínsecas.

- Funciones numéricas. Truncamiento y redondeo

x	<code>aint(x)</code>	<code>anint(x)</code>	<code>int(x)</code>	<code>nint(x)</code>
± 5.0	± 5.0	± 5.0	± 5	± 5
± 5.1	± 5.0	± 5.0	± 5	± 5
± 5.2	± 5.0	± 5.0	± 5	± 5
± 5.3	± 5.0	± 5.0	± 5	± 5
± 5.4	± 5.0	± 5.0	± 5	± 5
± 5.5	± 5.0	± 6.0	± 5	± 6
± 5.6	± 5.0	± 6.0	± 5	± 6
± 5.7	± 5.0	± 6.0	± 5	± 6
± 5.8	± 5.0	± 6.0	± 5	± 6
± 5.9	± 5.0	± 6.0	± 5	± 6

Funciones intrínsecas.

- Funciones numéricas. Matemáticas

<code>sqrt(x)</code>	<code>log(x)</code>
<code>conjg(z)</code>	<code>log10(x)</code>
<code>max(x1, x2, ...)</code>	<code>cos(x)</code>
<code>min(x1, x2, ...)</code>	<code>sin(x)</code>
<code>exp(x)</code>	<code>tan(x)</code>

<code>acos(x)</code>	$0 \leq \arccos(x) \leq \pi$
<code>asin(x)</code>	$-\pi/2 \leq \arcsen(x) \leq \pi/2$
<code>atan(x)</code>	$-\pi/2 \leq arctg(x) \leq \pi/2$
<code>atan2(x, y)</code>	$-\pi < arctg(y/x) \leq \pi$

Funciones intrínsecas.

- Funciones numéricas. Matemáticas

- $\text{abs}(x)$
 - Si x es real $\text{abs}(x) = \begin{cases} x & \text{si } x \geq 0 \\ -x & \text{si } x < 0 \end{cases}$

- Si x es complejo $\text{abs}(x) = \sqrt{X_r^2 + X_i^2}$

- $\text{mod}(x, y)$

- Ambos x e y pueden ser reales o enteros

$$\text{mod}(x, y) = x - \text{int}(x/y) \cdot y$$

Funciones intrínsecas.

- Funciones numéricas. Matemáticas.
- `epsilon(x)`
 - Devuelve un número del mismo tipo y kind que `x`, que es despreciable frente a la unidad.

$$1.d0 \sim 1.d0 \pm \text{epsilon}(x)$$

Funciones intrínsecas.

- Funciones para arrays. Búsqueda
 - `size(A,[dim=i])` (Los corchetes significan opcional)
 - Devuelve el tamaño de la dimensión i del array A
 - Ej.- `size(V)` devuelve el número de elementos del vector V
 - Ej.- `size(Matriz,dim=1)` devuelve el número de filas del array $Matriz$
 - Ej.- `size(Matriz,dim=2)` devuelve el número de columnas del array $Matriz$

Funciones intrínsecas.

- Funciones para arrays. Búsqueda
 - `maxval(Array)/minval(Array)`
 - Devuelve el mayor/menor elemento del array.
 - `maxval(Matriz,dim = i)/minval(Matriz,dim = i)`
 - `i=1` Devuelve un vector cuyos elementos son el máximo de cada una de las columnas.
 - `i=2` Devuelve un vector cuyos elementos son el máximo de cada una de las filas.
 - Ej.-
 - $A = \begin{pmatrix} 3 & 2 & 1 \\ 4 & -1 & 6 \end{pmatrix}$ `maxval(A,dim = 2) ! (3,6)`

Funciones intrínsecas.

- Funciones para arrays. Búsqueda
 - `maxloc(Array)/minloc(Array)`
 - Devuelve la posición (no el índice – empieza en 1) del mayor/menor elemento del array.
 - Ej. $V = (3 \quad -1 \quad 2)$ `minloc(V) ! 2`
 - Ej. $A = \begin{pmatrix} 3 & 2 & 1 \\ 4 & -1 & 6 \end{pmatrix}$ `maxloc(A) ! (2,3)`
 - Ej. $A = \begin{pmatrix} 3 & 2 & -1 \\ 4 & -1 & 6 \end{pmatrix}$ `minloc(A) ! (2,2)`

Funciones intrínsecas.

- Funciones para arrays. Matemáticas
 - `transpose(A)`
 - Devuelve la matriz transpuesta.

- Ej.- $A = \begin{pmatrix} 3 & 2 & 1 \\ 4 & -1 & 6 \end{pmatrix}$

- $B = \text{transpose}(A)$ $B = \begin{pmatrix} 3 & 4 \\ 2 & -1 \\ -1 & 6 \end{pmatrix}$

Funciones intrínsecas.

- Funciones para arrays. Matemáticas
 - `dot_product(vector_1,vector_2)`
 - Devuelve el producto escalar de los dos vectores.
 - `matmul(A,B)`
 - Devuelve el producto de dos matrices(arrays) con formas compatibles.

Funciones intrínsecas.

```
program producto
```

```
integer      :: V(3), U(3)
```

```
integer      :: A(3,2), B(2,3), C(3,3)
```

```
integer      :: i,j
```

```
A = 0
```

```
B = 0
```

```
do i = 1,2
```

```
    A(i,i) = 1
```

```
        do j = 1,3
```

```
            B(i,j) = j
```

```
        enddo
```

```
enddo
```

```
C = matmul(A,B)
```

```
do j = 1,3
```

```
    write(*,*) C(j,:)
```

```
enddo
```

```
end program producto
```



Funciones intrínsecas.

```
program producto
```

```
integer      :: V(3), U(3)
```

```
integer      :: A(3,2), B(2,3), C(3,3)
```

```
integer      :: i,j
```

```
A = 0
```

```
B = 0
```

```
do i = 1,2
```

```
    A(i,i) = 1
```

```
    do j = 1,3
```

```
        B(i,j) = j
```

```
    enddo
```

```
enddo
```

```
C = matmul(A,B)
```

```
do j = 1,3
```

```
    write(*,*) C(j,:)
```

```
enddo
```

```
end program producto
```

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$$

$$C = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 0 & 0 & 0 \end{pmatrix}$$

Funciones intrínsecas.

```
program producto

integer    :: V(2), U(3)
integer    :: A(3,2)
integer    :: i

do i =1,3
  U(i) = i
enddo

A = 1
V = matmul(U,A)

write(*,*) V

end program producto
```


Funciones intrínsecas.

```
program producto
```

```
integer    :: V(2), U(3)
```

```
integer    :: A(3,2)
```

```
integer    :: i
```

```
do i =1,3
```

```
    U(i) = i
```

```
enddo
```

```
A = 1
```

```
V = matmul(U,A)
```

```
write(*,*) V
```

```
end program producto
```

$$U = (1 \quad 2 \quad 3) \quad A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$$V = (6 \quad 6)$$

Funciones intrínsecas.

- Funciones para Character.
 - `len(string)`
 - Devuelve la longitud del string (cadena).
 - Ej.- `len(' Fortran & magic ') = 20`
 - `trim(string)`
 - Devuelve el mismo string quitando los espacios en blanco a la derecha del mismo.
 - Ej.- `trim (' Fortran & magic ') =
 ' Fortran & magic'`