

INSTRUCCIONES DE SALIDA Y DE ENTRADA

disp: Muestra un texto o un array

Sintaxis:

disp(X) %Se muestra el array X, sin mostrar su nombre. Si X contiene un texto, éste se muestra.

input: Solicita una entrada al usuario

Sintaxis:

Variable = input('mensaje') %Muestra el mensaje y lee un dato que se asigna a *Variable*.

Variable = input('mensaje', 's') %Recibe como dato un *texto* o *string* y se asigna a *Variable*.

ESTRUCTURAS DE CONTROL DE PROGRAMACIÓN

Consideramos 3 estructuras de control básicas:

- sentencia **if**
- lazo **for**
- lazo **while**

if

La sentencia **if** evalúa una expresión lógica y cuando la expresión es *verdadera*, ejecuta un grupo de sentencias. Las alternativas **elseif** y **else** proporcionan la ejecución de un grupo alternativo de sentencias. La palabra **end**, termina el último grupo de sentencias.

Recordemos que en las condiciones, el valor 0 es *falso* y cualquier otro valor es considerado *verdadero*.

Por ejemplo, las siguientes sentencias determinan si el año n, es un año bisiesto o no.

n= input('Ingrese al año n = ');	elseif rem(n,100) == 0
if rem(n,4) ~= 0	disp('No Bisiesto');
disp('No Bisiesto');	else
elseif rem(n,400) == 0	disp('Bisiesto');
disp('Bisiesto');	end;

Es importante entender cómo trabajan los operadores relacionales con las matrices.

Cuando se chequea la igualdad entre 2 variables, se podría usar

if (A == B) % El resultado es otra matriz de 0s y 1s

La forma apropiada de chequear la igualdad entre 2 variables matriz es usar la función *isequal*,

if isequal(A,B)....

for

El lazo **for** repite un grupo de sentencias un número fijo predeterminado de veces. Debe tener un **end**

```
for n = 1:10
    t(n) = n*n;
end
t
```

Es una buena idea indentar (separarlos con sangría) los lazos para mejor lectura.

```
for i = 1:2
    for j = 1:3
        H(i,j) = 1/(i+j);
    end
end
```

while

El lazo **while** repite un grupo de sentencias un número indefinido de veces, bajo el control de una condición lógica. El **while** viene acompañado de la palabra **end**.

Para realizar lo mismo que el 1er. ejemplo del **for**:

```
n=1;                    % también es posible t=[ ] ;
while (n<= 10)
    t(n) = n*n;            % también es posible t = [t n] ;
    n=n+1;
end;
t
```