Chapter Three
Loops and Controlling Command

Table 6 : Relational and logical operators

Operator	Description
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
~=	Not equal to
&	AND operator
|	OR operator
~	NOT operator

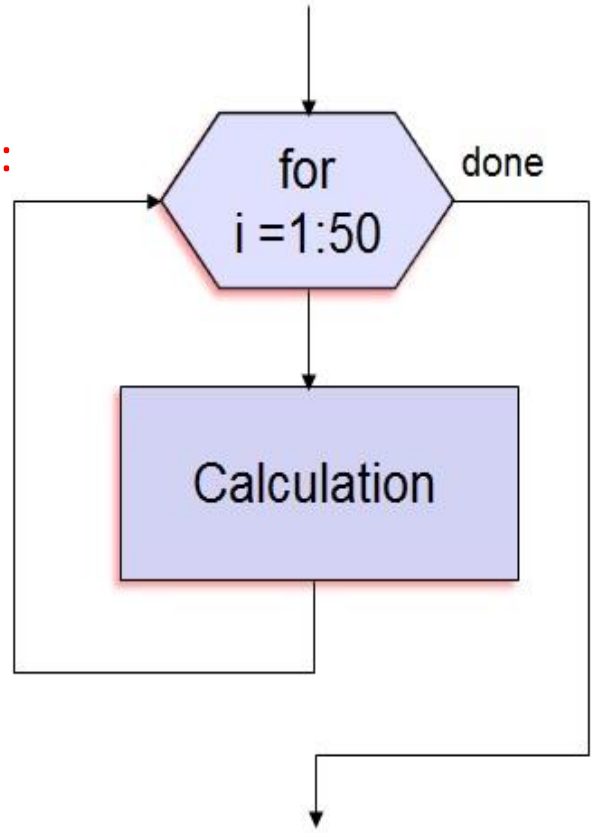
Loops

1. The " for ... end " loop

```
for  variable = expression
     statements
end
```

A simple example of for loop is:

```
for i = 1 : 50
     x = i*i
 end
```




```
for i=1:50
   calculation ;
end
```

The following statements form the 5-by-5 symmetric matrix A with (*i;j*) element *i / j* for *j* ≥ *i* :

```
n  = 5 ; A =  eye (n) ;
for j = 2 : n
      for  i = 1 : j-1
           A(i,j) = i / j ;  A(j,i) = i / j ;
       end
end
```

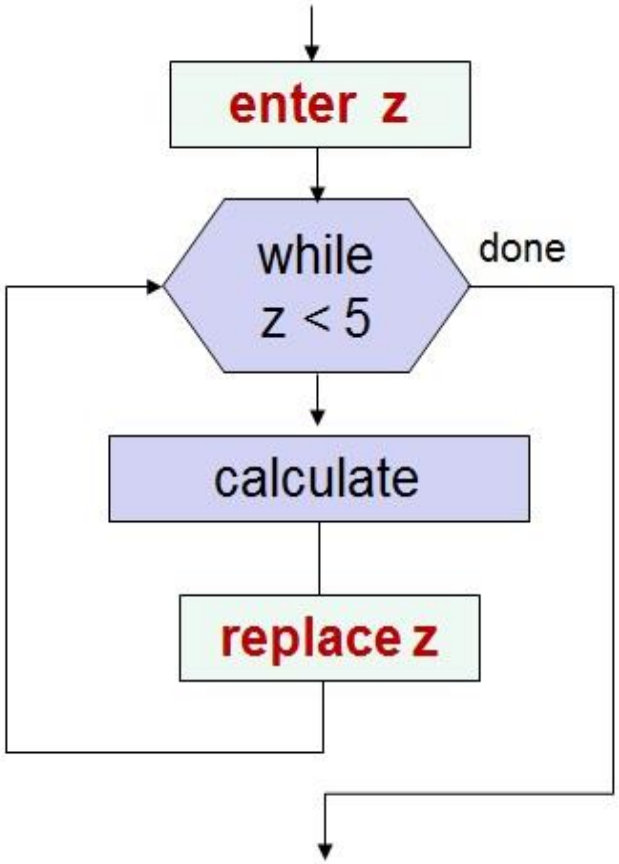
eye is used to Identity matrix

Syntax: Y = eye(n)

Description: Y = eye(n) returns the n-by-n identity matrix.

2. The " while ... end " loop

```
while expression
        statements
end
```




```
z=0;
while z<5
   calculate;
  z=z+1;
end
```

Ex:

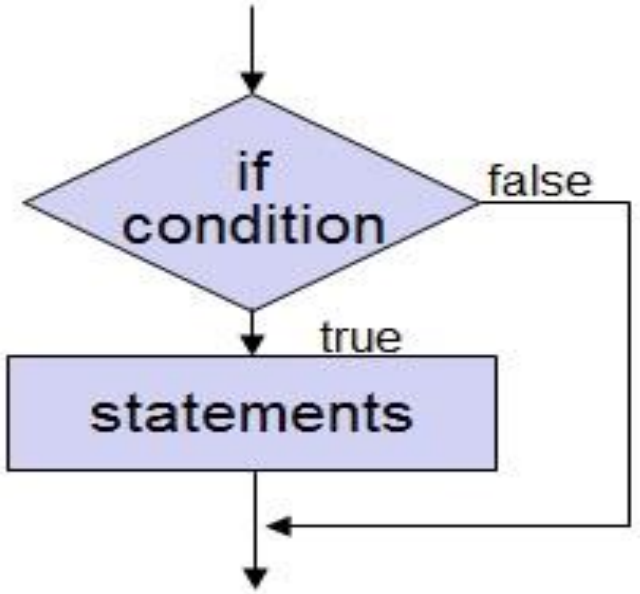
```
x=1;
while x <= 10
        x = 3*x
 end
```

Controlling Command

1. The "if...end" Structure

MATLAB supports the variants of "if" construct.

- if...end
- if...else...end
- if...elseif...else...end



The simplest form of the if statement is

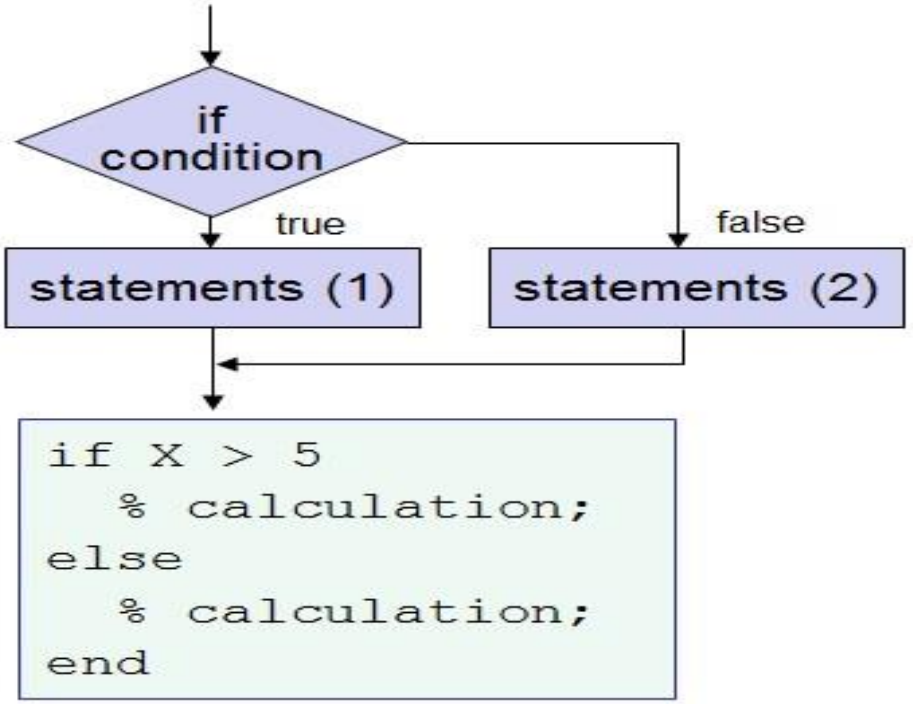
if expression
statements
end

```
if X > 5
   % calculation ;
end
```

Examples

1)
```
total  = b * b-4 * a * c ;
if total < 0
    disp ('total is negative value');
end
```

2)
```
total  = b * b-4 * a * c ;
if total < 0
    disp ('total is negative value');
else
    disp ('total is positive value')
end
```

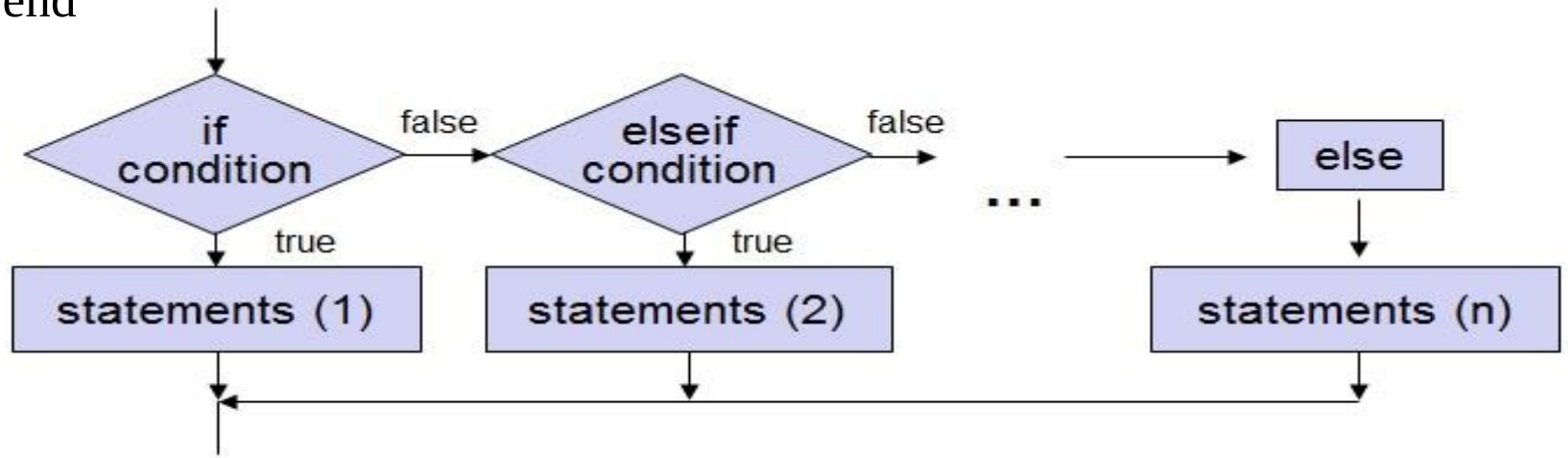


disp is used to display text or array

Syntax: disp(x)

Description: disp(x) displays an array, without printing the array name. If x contains a text string, the string is displayed. Another way to display an array on the screen is to type its name, but this prints a leading "x=," which is not always desirable. Note that disp does not display empty arrays.

3)
```
total  = b * b-4 * a * c ;
if total < 0
    disp ('total is negative value');
elseif total == 0
    disp ('total is zero value')
else
    disp ('total is positive value')
end
```




```
if A>5
   calculation;
elseif A<5
   calculation;
else
   calculation
end
```

Notes:

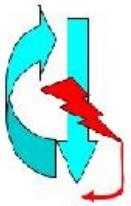
- **elseif** has no space between else and if (one word).
- No semicolon (**;**) is needed at the end of lines containing **if, else, end**.
- Indentation of if block is not required, but facilitate the reading.

Return

```
a = 0.5;
if    a < 1
    disp('Wrong parameters');
return
end
```

Continue

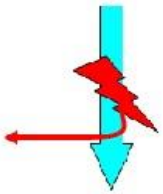
```
for  n = 1:50
if mod(n,7)
    continue
end
   disp(['Divisible by 7: ' num2str(n)])
end
```



- **break** – immediately breaks the loop
  - Breaks only one loop

- **continue** – jump to the end statement.
  - Does not break the loop

Only inside loops



- **return** – returns control to the command line (or to the calling function).

Anywhere

Switch

```
switch expiration
        case1
                statement,
        case2
                statement,
        otherwise
                statement,
end
```

Example: Write a code to convert x (cm) to any length untie type (in,ft, m, cm, and mm).

```
switch units
        case {'inch','in'}  % ‘units’ contains type of
    y = x*2.54;      % input, output is in cm
        case {'feet','ft'}
    y = x*2.54*12;
        case {'meter','m'}
    y = x*100;
        case {'centimeter','cm'}
    y = x;
        case {'millimeter','mm'}
    y = x/10;
        otherwise
      disp(['Unknown Units: ' units])
       y = NaN;
end
```