MATLAB has **four** flow control and/or branching instructions: for loops, while loops, if-else branching tests, and switch branching tests. All of these instructions end with an end statement, and it is frequently difficult to determine the extent of these instructions. Thus, it is very important to use indentation to indicate the structure of a code.

The general form of the **for** loop is

```
» for <variable> = <expression>
          <statement>
          ...
          <statement>
   end
```

where the variable is often called the index of the loop. The elements of the row vector <expression> are stored one at a time in the variable and then the statements up to the end statement are executed. For example, you can define the vector $x \in \mathbb{R}^n$ where $x_i = i \sin(i^2 \pi / n)$ by

```
» x = zeros(n, 1);
» for i = 1:n
          x(i) = i * sin( i^2 *π/n );
  end
```

Another example of the use of a for loop is the generation of the Hilbert matrix of order *n*.

```
» H = zeros(n);
» for i = 1:n
      »for j = 1:n
           H(i,j) = 1/(i+j-1);
       end
  end
```

Programing in MATLAB
Flow Control: the while loop

The second MATLAB loop structure is the while statement. The general form of the while loop is

```
» while <logical expression>
      <statement>
      ...
      <statement>
  end
```

where the statements are executed repeatedly as long as the <logical expression> is true.
For example, eps can be calculated by

```
» eps = 1;
» while 1+ eps > 1
      eps = eps/2;
  end
» eps = 2*eps
```

Programing in MATLAB

Flow Control: the if statement

The simplest form of the if statement is

```
» if <logical expression>
      <statement>
      ...
      <statement>
  end
```

where the statements are evaluated as long as the <logical expression> is true. The <logical expression> is generally of the form <arithmetic expression-left> rop <arithmetic expression-right> where rop is one of the relational operators shown below:

< Less than.
<= Less than or equal to.
== Equal.
> Greater than.
>= Greater than or equal to.
~= Not equal to.
strcmp(a, b) Compares strings.

Programing in MATLAB

Flow Control: the if statement

A second form of the if statement is

```
» if <logical expression>
      <statement group 1>
  else
      <statement group 2>
  end
```

where statement group 1 is evaluated if the <logical expression> is true and statement group 2 is evaluated if it is false.

Programing in MATLAB

Flow Control: the if statement

The final form of the if statement is

```
» if <logical expression 1>
      <statement group 1>
  elseif <logical expression 2>
      <statement group 2>
  ...
  elseif <logical expression r>
      <statement group r>
  else
      <statement group r+1>
  end
```

where statement group 1 is evaluated if the <logical expression 1> is true, statement group 2 is evaluated if the <logical expression 2> is true, etc. The final else statement is not required. If it occurs and if none of the logical expressions is true, statement group r+1 is evaluated. If it does not occur and if none of the logical expressions is true, then none of the statement groups are executed.

Programing in MATLAB

Flow Control: the switch command

The switch command executes particular statements based on the value of a variable or an expression. Its general form is

```
» switch <variable or expression>
      case <Value 1>,
            <statement group 1>
      case { <Value 2a>, <Value 2b>, <Value 2c>, ..., <Value 2m>} ,
            <statement group 2>
      ...
      case <value n>,
            <statement group r>
      otherwise,
            <statement group r+1>
  end
```

where statement group 1 is evaluated if the variable or expression has <Value 1>, where statement group 2 is evaluated if the variable or expression has values <Value 2a> or <Value 2b> or <Value 2c>, etc.

Programing in MATLAB
Flow Control: the switch command

Note that if a case has more than one value, then all the values must be surrounded by curly brackets. The final otherwise is not required. If it occurs and if none of the values match the variable or expression, then statement group r+1 is evaluated. If it does not occur and if none of the values match, then none of the statement groups are executed.

Here is a code that finds the number of days in a month.

```
» switch  month
     case {1, 3, 5, 7, 8, 10, 12}
          days=31;
     case {4, 6, 9, 11}
          days=30;
     case 2
          days=28;
  end
```