

The script file (saved as Exp6_1) is executed in the Command Window:

```
>> Exp6_1
NdaysTabove75 =      For 7 days the temp was above 75.
      7
NdaysTbetween65and80 =  For 12 days the temp was between 65 and 80.
      12
datesTbetween50and60 =      Dates of the month with
      1      4      5      7      21      23      temp between 50 and 60.
```

6.2 CONDITIONAL STATEMENTS

A conditional statement is a command that allows MATLAB to make a decision of whether to execute a group of commands that follow the conditional statement, or to skip these commands. In a conditional statement, a conditional expression is stated. If the expression is true, a group of commands that follow the statement are executed. If the expression is false, the computer skips the group. The basic form of a conditional statement is:

```
if conditional expression consisting of relational and/or logical operators.
```

Examples:

```
if a < b
if c >= 5
if a == b
if a ~= 0
if (d<h) & (x>7)
if (x~=13) | (y<0)
```

All the variables must
have assigned values.

- Conditional statements can be a part of a program written in a script file or a user-defined function (Chapter 7).
- As shown below, for every `if` statement there is an `end` statement.

The `if` statement is commonly used in three structures, `if-end`, `if-else-end`, and `if-elseif-else-end`, which are described next.

6.2.1 The `if-end` Structure

The `if-end` conditional statement is shown schematically in Figure 6-1. The figure shows how the commands are typed in the program, and a flowchart that symbolically shows the flow, or the sequence, in which the commands are executed. As the program executes, it reaches the `if` statement. If the conditional expression in the `if` statement is true (1), the program continues to execute the

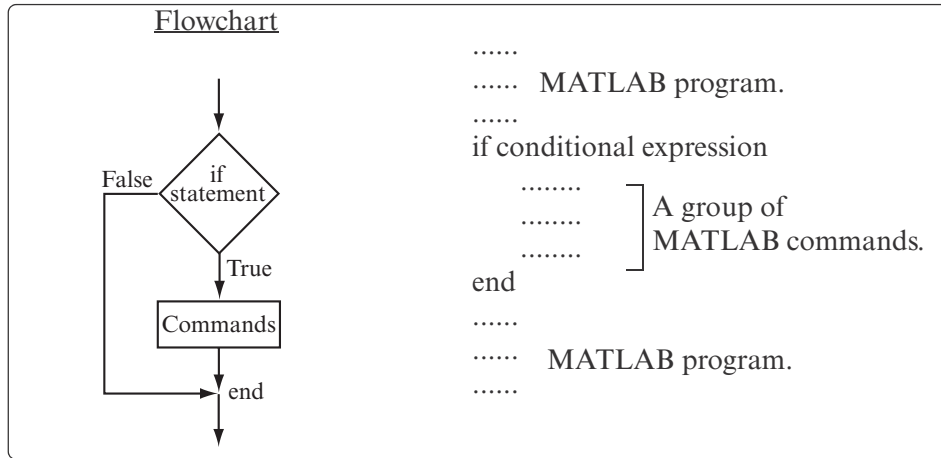


Figure 6-1: The structure of the if-end conditional statement.

commands that follow the `if` statement all the way down to the `end` statement. If the conditional expression is false (0), the program skips the group of commands between the `if` and the `end`, and continues with the commands that follow the `end`.

The words `if` and `end` appear on the screen in blue, and the commands between the `if` statement and the `end` statement are automatically indented (they don't have to be), which makes the program easier to read. An example where the `if-end` statement is used in a script file is shown in Sample Problem 6-2.

Sample Problem 6-2: Calculating worker's pay

A worker is paid according to his hourly wage up to 40 hours, and 50% more for overtime. Write a program in a script file that calculates the pay to a worker. The program asks the user to enter the number of hours and the hourly wage. The program then displays the pay.

Solution

The program in a script file is shown below. The program first calculates the pay by multiplying the number of hours by the hourly wage. Then an `if` statement checks whether the number of hours is greater than 40. If so, the next line is executed and the extra pay for the hours above 40 is added. If not, the program skips to the `end`.

```

t=input('Please enter the number of hours worked ');
h=input('Please enter the hourly wage in $ ');
Pay=t*h;
if t>40

```