Example 5.1

The program in Fig.5.3 reads four values a, b, c, and d from the terminal and evaluates the ratio of (a+b) to (c-d) and prints the result, if c-d is not equal to zero.

The program given in Fig.5.3 has been run for two sets of data to see that the paths function properly. The result of the first run is printed as

Ratio = -3.181818

ILLUSTRATION OF ***if*** STATEMENT

Program

```
main()
{
    int a, b, c, d;
    float ratio;

    printf("Enter four integer values\n");
    scanf("%d %d %d %d", &a, &b, &c, &d);

    if (c-d  != 0)  /* Execute statement block */
    {
        ratio = (float)(a+b)/(float)(c-d);
        printf("Ratio = %f\n", ratio);
    }
}
```

Output

```
Enter four integer values
12  23  34  45
Ratio = -3.181818

Enter four integer values
12  23  34  34
```

Fig. 5.3 *Illustration of simple* ***if*** *statement*

Example 5.2

The program in Fig.5.4 counts the number of boys whose weight is less than 50 kgs and height is greater than 170 cm.

The program has to test two conditions, one for weight and another for height. This is done using the compound relation

```
if (weight < 50 && height > 170)
```

This would have been equivalently done using two **if** statements as follows:

```
if  (weight < 50)
   if  (height > 170)
      count = count +1;
```

If the value of **weight** is less than 50, then the following statement is executed, which in turn is another **if** statement. This **if** statement tests **height** and if the **height** is greater than 170, then the **count** is incremented by 1.

COUNTING WITH **if**

Program

```
main()
{
    int count, i;
    float weight, height;

    count = 0;
    printf("Enter weight and height for 10 boys\n");

    for (i =1; i <= 10; i++)
    {
       scanf("%f %f", &weight, &height);
       if (weight < 50 && height > 170)
          count = count + 1;
    }

    printf("Number of boys with weight < 50 kgs\n");
    printf("and height > 170 cm = %d\n", count);
}
```

Output

```
Enter weight and height for 10 boys
45  176.5
55  174.2
47  168.0
49  170.7
54  169.0
53  170.5
49  167.0
48  175.0
47  167
51  170
Number of boys with weight < 50 kgs
and height > 170 cm = 3
```

***Fig. 5.4** Use of **if** for counting*

Example 5.3

A program to evaluate the power series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots\ldots + \frac{x^n}{n!}, \quad 0 < x < 1$$

is given in Fig. 5.6. It uses **if.....else** to test the accuracy.

The power series contains the recurrence relationship of the type

$$T_n = T_{n-1} \left(\frac{x}{n}\right) \text{ for } n > 1$$

$$T_1 = x \qquad \text{for } n = 1$$

$$T_0 = 1$$

If T_{n-1} (usually known as *previous term*) is known, then T_n (known as *present term*) can be easily found by multiplying the previous term by x/n. Then

$$e^x = T_0 + T_1 + T_2 + \ldots\ldots + T_n = \text{sum}$$

EXPERIMENT WITH **if...else** STATEMENT

Program

```
#define ACCURACY 0.0001

main()
{
    int n, count;
    float x, term, sum;

    printf("Enter value of x:");
    scanf("%f", &x);

    n = term = sum = count = 1;

    while (n <= 100)
    {
       term = term * x/n;
       sum = sum + term;
       count = count + 1;
       if (term < ACCURACY)
          n = 999;
       else
          n = n + 1;
    }

    printf("Terms = %d Sum = %f\n", count, sum);

}
```

Output

```
Enter value of x:0
Terms = 2 Sum = 1.000000
```

```
Enter value of x:0.1
Terms = 5 Sum = 1.105171

Enter value of x:0.5
Terms = 7 Sum = 1.648720

Enter value of x:0.75
Terms = 8 Sum = 2.116997

Enter value of x:0.99
Terms = 9 Sum = 2.691232

Enter value of x:1
Terms = 9 Sum = 2.718279
```

Fig 5.6 *Illustration of* ***if...else*** *statement*

Example 5.4

The program in Fig. 5.8 selects and prints the largest of the three numbers using nested **if....else** statements.

SELECTING THE LARGEST OF THREE VALUES

Program

```
main()
{
    float A, B, C;

    printf("Enter three values\n");
    scanf("%f %f %f", &A, &B, &C);

    printf("\nLargest value is  ");

    if (A>B)
    {
        if (A>C)
           printf("%f\n", A);
        else
           printf("%f\n", C);
    }
    else
    {
        if (C>B)
           printf("%f\n", C);
       else
           printf("%f\n", B);
```

```
    }
}
```

Output

```
Enter three values
23445  67379  88843

Largest value is  88843.000000
```

Fig 5.8 *Selecting the largest of three numbers*

Example 5.5

An electric power distribution company charges its domestic consumers as follows:

Consumption Units	*Rate of Charge*
0 - 200	Rs. 0.50 per unit
201 - 400	Rs. 100 plus Rs.0.65 per unit excess of 200
401 - 600	Rs. 230 plus Rs.0.80 per unit excess of 400
601 and above	Rs. 390 plus Rs.1.00 per unit excess of 600

The program in Fig.5.10 reads the customer number and power consumed and prints the amount to be paid by the customer.

USE OF **else if** LADDER

Program

```
main()
{
    int  units, custnum;
    float charges;

    printf("Enter CUSTOMER NO. and UNITS consumed\n");
    scanf("%d %d", &custnum, &units);

    if (units <= 200)
       charges = 0.5 * units;
    else if (units <= 400)
        charges = 100 + 0.65 * (units - 200);
     else if (units <= 600)
                 charges = 230 + 0.8 * (units - 400);
              else
                 charges = 390 + (units - 600);

    printf("\n\nCustomer No: %d: Charges = %.2f\n",
           custnum, charges);
}
```

Output

```
Enter CUSTOMER NO. and UNITS consumed 101  150
Customer No:101 Charges = 75.00

Enter CUSTOMER NO. and UNITS consumed 202  225
Customer No:202 Charges = 116.25

Enter CUSTOMER NO. and UNITS consumed 303  375
Customer No:303 Charges = 213.75

Enter CUSTOMER NO. and UNITS consumed 404  520
Customer No:404 Charges = 326.00

Enter CUSTOMER NO. and UNITS consumed 505  625
Customer No:505 Charges = 415.00
```

Fig. 5.10 *Illustration of* ***else..if*** *ladder*

Example 5.6

An employee can apply for a loan at the beginning of every six months, but he will be sanctioned the amount according to the following company rules:

Rule 1 : An employee cannot enjoy more than two loans at any point of time.
Rule 2 : Maximum permissible total loan is limited and depends upon the category of the employee.

A program to process loan applications and to sanction loans is given in Fig. 5.12.

Program

```
#define   MAXLOAN  50000

main()
{
    long int loan1, loan2, loan3, sancloan, sum23;

    printf("Enter the values of previous two loans:\n");
    scanf(" %ld %ld", &loan1, &loan2);

    printf("\nEnter the value of new loan:\n");
    scanf(" %ld", &loan3);

    sum23 = loan2 + loan3;
    sancloan = (loan1>0)? 0 : ((sum23>MAXLOAN)?
                MAXLOAN - loan2 : loan3);

    printf("\n\n");
    printf("Previous loans pending:\n%ld %ld\n",loan1,loan2);
```

```
        printf("Loan requested  = %ld\n", loan3);
        printf("Loan sanctioned = %ld\n", sancloan);

    }
```

Output

```
Enter the values of previous two loans:
0    20000

Enter the value of new loan:
45000

Previous loans pending:
0   20000
Loan requested  = 45000
Loan sanctioned = 30000

Enter the values of previous two loans:
1000   15000

 Enter the value of new loan:
25000

Previous loans pending:
1000   15000
Loan requested  = 25000
Loan sanctioned = 0
```

Fig 5.12 *Illustration of the conditional operator*

Example 5.7

Program presented in Fig.5.13 illustrates the use of the **goto** statement.

The program evaluates the square root for five numbers. The variable count keeps the count of numbers read. When count is less than or equal to 5, **goto read**; directs the control to the label **read**; otherwise, the program prints a message and stops.

*USE OF **goto** STATEMENT*

Program

```
#include  <math.h>
main()
{
     double x, y;
     int count;
```

```
        count = 1;

        printf("Enter FIVE real values in a LINE \n");
    read:
        scanf("%lf", &x);
        printf("\n");
        if (x < 0)
           printf("Value - %d is negative\n",count);
        else
        {
           y = sqrt(x);
           printf("%lf\t %lf\n", x, y);
        }
        count = count + 1;

        if (count <= 5)
    goto read;
        printf("\nEnd of computation");
    }
```

Output

```
Enter FIVE real values in a LINE
50.70  40  -36  75  11.25
50.750000        7.123903
40.000000        6.324555

Value -3 is negative
75.000000        8.660254
11.250000        3.354102
End of computation
```

Fig.5.13 *Use of the* ***goto*** *statement*