Subroutines are used when it is desired to return to the main program more than one value of a variable or more than one variable. The basics of a subroutine are very similar to the function subprogram. There are two main differences.

1. There is no value associated with the name of a subroutine which means there is no significance to the first letter in the name. Otherwise the naming of a subroutine follows the procedures for the function.

2. A subroutine is called from the main program with a CALL statement. The format for the CALL is as follows:

```
CALL STAT(list)
```

STAT is the name of the subroutine. The list contains the names of variables that are passed both from and to the main program. As many variables as the programmer desires may be passed.

The first line of the subroutine contains the name of the subroutine and the word SUBROUTINE.

```
SUBROUTINE STAT(list)
```

Again the list represents the variables that are to be passed from or to the main program. The lists in the CALL and SUBROUTINE statements need to contain the same amount of variables but they do not have to have the same name. For example,

```
CALL STAT(M,Y,AVE,STDEV)
SUBROUTINE STAT(N,X,AVE,STDV)
```

The variables do have to match up, i.e. the Y passed from the main program becomes X in the subroutine. Likewise when X is passed from the subroutine to the main program, Y assumes the value that X had. The subroutine must end with an END statement and it must contain at least one RETURN statement. RETURN often occurs immediately before the END, but it does not have to. The following subroutine contains the calculation procedure for the average and standard deviation of an array of numbers.

```
SUBROUTINE STAT(N, X, AVE, STDV)
  DIMENSION X(25)
  SUM = 0.0
  DO 10 I = 1, N
```

SUBROUTINE STAT(N, X, AVX, STDV)
DIMENSION X(N)
AVX = AVE(X, N)
SUM = 0.0
DO 10 I = 1, N
   SUM = SUM + (X(I) – AVX)**2
10 CONTINUE
STDV = SQRT(SUM/(N – 1))
RETURN
END

You could call either of the subroutines by the following statements which are part of a main program.

PROGRAM CALC1
DIMENSION Y(25)
OPEN (UNIT = 12, FILE = ‘NUM.DAT’, STATUS = ‘OLD’) READ (12, *) M
DO 10 I = 1, M
   READ (12, *) Y(I)
10 CONTINUE
CLOSE (UNIT = 12)
CALL STAT(M, Y, AVE, STDEV)
PRINT*, ‘THE AVERAGE IS = ‘, AV
PRINT*, ‘THE STANDARD DEVIATION = ‘, STDEV
The subroutine STAT receives input data from the main program through the CALL statement. The array called Y and one other variable, M, are sent to the subroutine. In the case of the first SUBROUTINE, the computed average, AVE, and standard deviation, STDV, are passed back to the main program, again by means of the CALL statement. For the second SUBROUTINE, the computed average is stored in AVX, and the standard deviation in STDV, and are passed back to the main program by means of the CALL statement.

As was discussed earlier, the main program and the subroutine are separate program entities. All variables, statement numbers, and DIMENSION statements in the subroutine are independent from those in the main program. That is why a DIMENSION statement appears in the subroutine as well as in the main program.

To summarize, note the following conclusions regarding SUBROUTINES.

1. The names in the argument list in the main program do not need to be the same as those in the argument list of the SUBROUTINE, but they must be in the same order and of the same type. Just as in the case of FUNCTION subprograms.
2. The only way SUBROUTINES communicate with the main program is through the argument list.
3. You can change the variables in the list as well as calculate new ones.
4. The name of the SUBROUTINE is not a variable as it is in the case of FUNCTION subprograms so you don’t need to specifically make it real or integer.
5. You can call other SUBROUTINES or FUNCTIONS from a SUBROUTINE and you can call SUBROUTINES from FUNCTIONS.
6. The RETURN in a subprogram tells the computer to return to the line after the call is made in the calling program. Sometimes you will have several returns in a subprogram. This is usually in conjunction with some if statements. Just remember that when program control comes to a RETURN, it will return to the calling program.
7. You may PRINT or WRITE or READ from a SUBROUTINE.