

from different places in your program, and RETURN to the appropriate command each time. The following example uses the TRACE feature described in Section 7 to demonstrate the flow of control with GOSUB and RETURN commands.

```

NEW
OK
10 TRACE
20 PRINT "BEGINNING";
30 GOSUB 100
40 PRINT "MIDDLE";
50 GOSUB 100
60 PRINT "END";
70 UNTRACE
80 END
100 PRINT "SUBROUTINE";
110 RETURN
RUN
[20 PRINT "BEGINNING";]      BEGINNING
[30 GOSUB 100]
[100 PRINT "SUBROUTINE";]    SUBROUTINE
[110 RETURN]
[40 PRINT "MIDDLE";]        MIDDLE
[50 GOSUB 100]
[100 PRINT "SUBROUTINE";]    SUBROUTINE
[110 RETURN]
[60 PRINT "END";]           END
[70 UNTRACE]
OK

```

After executing the TRACE command in line 10, XYBASIC prints the bracketed line number and contents of each line it executes. You can see that executing line 110 RETURNS control first to line 40 and then to line 60.

If your program executes a RETURN without a corresponding GOSUB, an RG (Return without Gosub) error will occur, as XYBASIC does not know where to return. If your program GOSUBs to a nonexistent line number, a US (Undefined Statement) error will occur.

Each time XYBASIC executes a GOSUB it uses memory space to store information about where to RETURN, and the used space is reclaimed when the corresponding RETURN is executed. If insufficient memory space remains, an OM (Out of Memory) error will occur. The FRE example in Section 3 below demonstrates how memory space is used and later reclaimed.

The following example uses GOSUB and RETURN to calculate the greatest common divisor (G.C.D.) of three numbers, i.e. the largest number which divides each with no remainder. First it finds the G.C.D. of X and Y, then finds the G.C.D. of that value and Z; the result is easily shown to be the G.C.D. of X, Y and Z.