Since XYBASIC tests the condition specified by each interrupt before executing each command, interrupts slow down XYBASIC considerably. You should notice that ENABLE interrupts are controlled by software, not using the hardware interrupt facilities of your computer. Of course you can use the CALL and SCALL commands to access machine language routines for activation and servicing of hardware interrupts.

DISABLE

You can deactivate all ENABLEd interrupts by just saying DISABLE. Alternatively, you can deactivate a specific interrupt:

```
DISABLE 10
```

disables only the interrupt which you ENABLEd in line 10. If you try to DISABLE a nonexistent interrupt, an EN error will occur. XYBASIC also disables all interrupts when you execute a RUN.

Example:
In the WAIT example of Section 9, Emil Post used the following program to warn him of the arrival of his boss:

```
NEW
10 WAIT 5, &11111, &11100000, $
20 PRINT "WAKE UP!  HERE HE COMES!"
```

Emil has now been given a XYBASIC program which plays blackjack, but he does not want his boss to catch him playing. He therefore uses ENABLE instead of WAIT, allowing him to play blackjack and still be warned when his boss arrives.

```
NEW
10 ENABLE 20, 5, &11111, &11100000, $
15 GOTO 100
20 PRINT "BET FAST!  HERE HE COMES!"
25 DISABLE 10
30 RETURN
100 'START OF BLACKJACK PROGRAM
...
```

The ENABLE command of line 10 specifies that an interrupt will occur when the value of any of bits 0 through 4 of input port 5 becomes 1. After the ENABLE the program plays blackjack with Emil. When his boss enters, one of bits of port 5 becomes 1 and an interrupt occurs, transferring control to the subroutine at line 20. A warning message is printed by line 20. Line 25 DISABLEs the interrupt, so that the message will not be repeated even though one of the bits of port 5 is still 1, and line 30 RETURNs control to the blackjack program at the point it was interrupted.