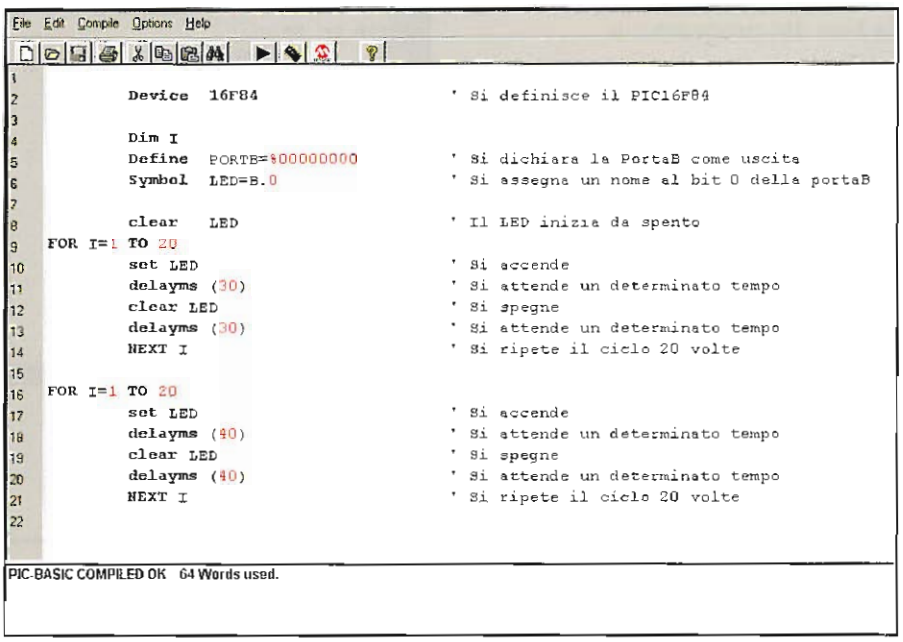
Fermare il tempo

Normalmente lo scopo di un programma è quello di realizzare una determinata funzione, a volte però, può essere necessario il contrario, cioè che si fermi e non faccia nulla. Le ragioni di tutto ciò possono essere molteplici e il LetPicBasic ci permette di farlo in diverse forme, a seconda del caso.



Il compilatore non genera errori se non utilizziamo l'istruzione END.

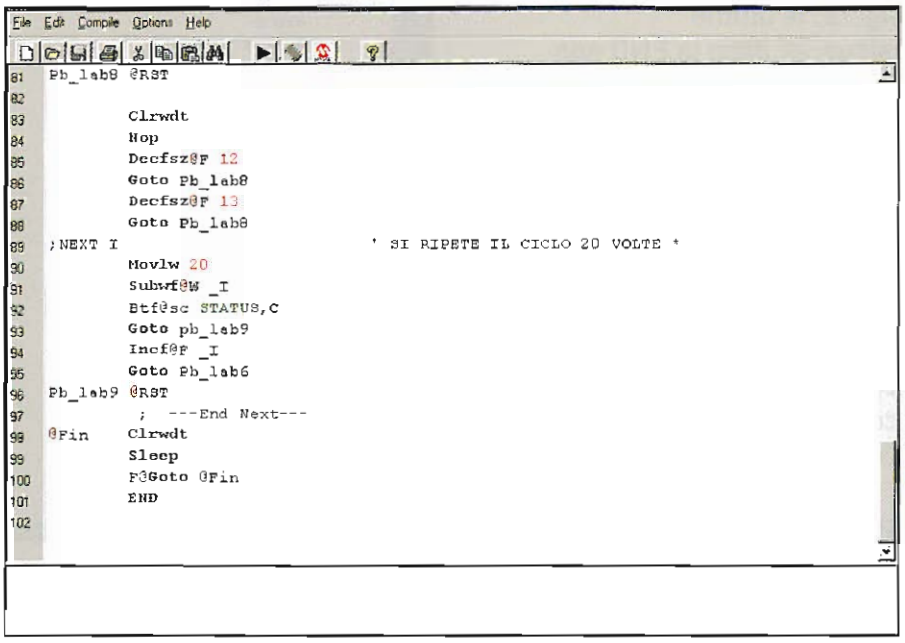
Arresto della compilazione del programma

Siamo abituati a usare il comando END in tutti i nostri programmi: tutti i programmi devono terminare con questa istruzione. Provate a realizzare un programma qualsiasi, ad esempio quello che noi riportiamo nella figura, però non scrivete alla fine l'istruzione END. Compilatelo.

Come potete osservare non si genera alcun errore.

Ora aprite lo stesso programma che avete appena terminato di eseguire, però solo la parte tradotta in codice assembler, cioè quella che ha estensione "asm".

Se andate alla fine delle istruzioni potrete verificare che è stata aggiunta una istruzione END.



Il LetPicBasic aggiunge l'istruzione END se noi non la mettiamo nel codice originale.

Il LetPicBasic si incarica di metterla per noi, perché il compilatore che trasforma il codice assembler in codice macchina esige che questa istruzione segni la fine del programma.

Tuttavia, noi non ci siamo mai chiesti perché succede questo, o che cos'è che fa esattamente questo comando. Dopo una sentenza END si ferma la compilazione del codice, in modo che tutto ciò che è scritto dopo di essa è come se non esistesse, né tanto meno si producono errori se aggiungiamo ulteriori istruzioni al programma. Nella figura seguente abbiamo aggiunto un ciclo simile a quelli precedenti dopo l'istruzione END e abbiamo compilato. Possiamo notare che, non solo non si sono prodotti errori, ma il numero delle linee compilate non è cambiato rispetto alla prima figura; le ultime istruzioni dopo la END non sono state compilate.

```
File  Edit  Compile  Options  Help
FOR I=1 TO 20
        set LED                 ' Si accende
        delayms (30)            ' Si attende un determinato tempo
        clear LED               ' Si spegne
        delayms (30)            ' Si attende un determinato tempo
        NEXT I                  ' Si ripete il ciclo 20 volte

FOR I=1 TO 20
        set LED                 ' Si accende
        delayms (40)            ' Si attende un determinato tempo
        clear LED               ' Si spegne
        delayms (40)            ' Si attende un determinato tempo
        NEXT I                  ' Si ripete il ciclo 20 volte
END

FOR I=1 TO 20
        set LED                 ' Si accende
        delayms (50)            ' Si attende un determinato tempo
        clear LED               ' Si spegne
        delayms (50)            ' Si attende un determinato tempo
        NEXT I                  ' Si ripete il ciclo 20 volte

PIC-BASIC COMPILED OK  64 Words used.
```

Dopo l'istruzione END non viene più compilata alcuna istruzione.

Arresto dell'esecuzione di un programma

Se con END si ferma la compilazione di un programma, con STOP si ferma l'esecuzione dello stesso.

Non bisogna confondere una cosa con l'altra, dato che nel secondo caso la compilazione si produce, ed è al momento di eseguire il programma che l'istruzione viene tenuta in considerazione.

Provate a realizzare, ad

```
File  Edit  Compile  Options  Help
        Device  16F84           ' Si definisce il PIC16F84

        Dim I
        Define  PORTB=%00000000 ' Si dichiara la PortaB come uscita
        Define  PORTA=%00000001 ' Si dichiara il bit 0 della PortaA
                                ' come ingresso
        Symbol  LED=B.0         ' Si assegna un nome al bit 0 della portaB
        clear   LED             ' Il LED inizia da spento
If PORTA&1=1 then goto LOOP30   ' Secondo il valore del bit 0 di PORTA
        goto LOOP40             ' si entra in un ciclo oppure un altro

LOOP30: FOR I=1 TO 20
        set LED                 ' Si accende
        delayms (30)            ' Si attende un determinato tempo
        clear LED               ' Si spegne
        delayms (30)            ' Si attende un determinato tempo
        NEXT I                  ' Si ripete il ciclo 20 volte
        STOP
LOOP40: FOR I=1 TO 20
        set LED                 ' Si accende
        delayms (40)            ' Si attende un determinato tempo
        clear LED               ' Si spegne
        delayms (40)            ' Si attende un determinato tempo
        NEXT I                  ' Si ripete il ciclo 20 volte
        STOP
END
```

Dopo qualsiasi istruzione di STOP si ferma l'esecuzione del programma.