

```
gtemp1.asm - Blocco note
File Modifica Cerca ?
;SET LED      SetBit PORTB,00
;DELAYMS (30) * SI ASPETTA PER UN DETERMINATO TEMPO *
    Movlw 30
    MovBwf 13
    C1rBf 12
Pb_1ab3 ORST
    Clrwdt
    Nop
    DecfszBf 12
    Goto Pb_1ab3
    DecfszBf 13
    Goto Pb_1ab3
;RESET (RST)
; Start of Inline Code
    Nop
    Nop
    Nop
; End of Inline Code
;CLEAR LED      ClrBit PORTB,00
;DELAYMS (30) * SI ASPETTA PER UN DETERMINATO TEMPO *
    Movlw 30
    MovBwf 13
    C1rBf 12
Pb_1ab4 ORST
    Clrwdt
    Nop
    DecfszBf 12
    Goto Pb_1ab4
```

Programma generato a partire dalle istruzioni con ASM... ENDASM.

l'estensione ASM. Se lo apriamo verificheremo come è stata prodotta la traduzione. Secondo se le istruzioni sono state introdotte direttamente o specificando che sono istruzioni assembler (con ASM...ENDASM) la traduzione può apparire differente. Tuttavia se apriamo i file HEX generati che sono quelli che passeranno al microcontroller, possiamo verificare che i valori sono identici in entrambi i casi.

Errori tipici

```
gtemp1b.asm - Blocco note
File Modifica Cerca ?
;DELAYMS (30) * SI ASPETTA PER UN DETERMINATO TEMPO *
    Movlw 30
    MovBwf 13
    C1rBf 12
Pb_1ab3 ORST
    Clrwdt
    Nop
    DecfszBf 12
    Goto Pb_1ab3
    DecfszBf 13
    Goto Pb_1ab3
; INIZIO IL CODICE ASSEMBLER
    Bcf PORTB,4
    Bcf PORTB,4
; FINE DEL CODICE ASSEMBLER
;CLEAR LED      ClrBit PORTB,00
;DELAYMS (30) * SI ASPETTA PER UN DETERMINATO TEMPO *
    Movlw 30
    MovBwf 13
    C1rBf 12
Pb_1ab4 ORST
    Clrwdt
    Nop
    DecfszBf 12
    Goto Pb_1ab4
    DecfszBf 13
    Goto Pb_1ab4
;NEXT 1
    Movlw 20
    SubwFw 1
```

Programma generato a partire da istruzioni introdotte direttamente.

Se qualche volta avete provato ad apportare modifiche agli esercizi che vi abbiamo proposto, invece di seguire passo a passo le nostre istruzioni, potrebbe esservi capitato di generare errori al momento della compilazione. Vediamo se si tratta dei seguenti. Se cercate di compilare il programma della prima figura della pagina successiva vedrete che non è possibile anche se a prima vista si tratta di un programma già visto, però con le istruzioni assembler inserite tra le direttive ASM... ENDASM. Che cosa succede?

```
Esemp1.hex - Blocco note
File Modifica Cerca ?
:10000000000000001686000012061001309500061416
:100010001E300D008C0164000000000000280D0002
:10002000028000000000000006101E300D000C011F
:10003000640000000C001820AD0018201430150252
:0E0040003182428950A07286400630024286A
:02400E00F13F80
:00000001FF
```

File HEX generato per questo esempio in entrambi i casi.

Per rispondere ci baseremo sulla differenza che abbiamo detto esistere tra farlo in un modo e farlo in un altro. Abbiamo già visto che alla fine il risultato è lo stesso, però in quest'ultimo esempio le istruzioni devono essere copiate tali e quali, per essere interpretate dall'assembler e i commenti introdotti dopo le istruzioni confondono l'assembler (nell'assembler i commenti sono preceduti da ";" al posto di ""). Provate con l'esercizio successivo e vedrete come si risolve il problema. L'introduzione indiretta

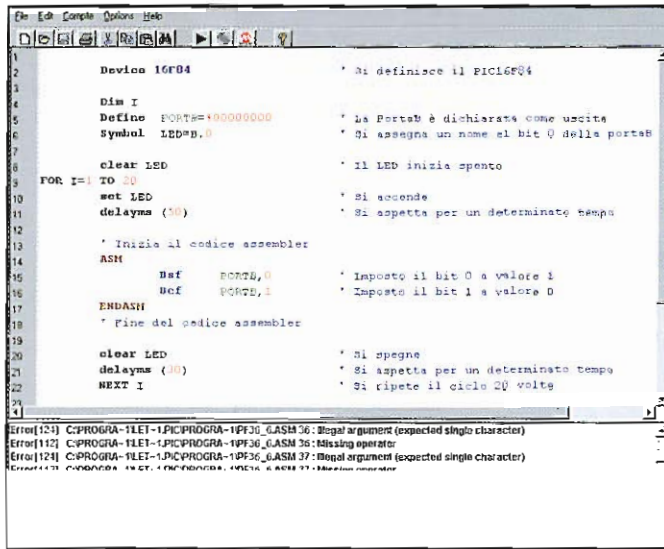
delle istruzioni assembler ha il vantaggio di permettere di usare le utilità proprie dell'assembler, come le macro, le altre direttive, le funzioni, ecc. che in altro modo non sarebbero accettate dal compilatore BASIC.

Per acquisire pratica

Vi proponiamo ora di fare voi da compilatore. Si tratta di un esercizio semplice anche se necessita di un po' di pazienza. Dopo aver aperto un esercizio .ASM di quelli generati dal programma LetPicBasic gli abbiamo detto di tradurlo in codice assembler. Ora le istruzioni che rimangono non sono ancora del tutto assembler, ad esempio tutte le istruzioni che hanno il carattere "@" come parte del nome, sono un passo intermedio prima dell'assembler. In realtà sono macro

che l'assembler tradurrà in sue proprie istruzioni grazie al file "14bit.inc" che viene incluso. Aprite questo file e il file generato .ASM e sostituite in quest'ultimo ogni istruzione con la @ con la sua corrispondente macro. Per esempio, MOV@FW verrà sostituita dalle tre linee che si trovano fra "macro reg" e "endm". Come potete vedere può succedere che dopo una sostituzione rimangano ancora altre @, però non preoccupatevi, fatelo passo a passo.

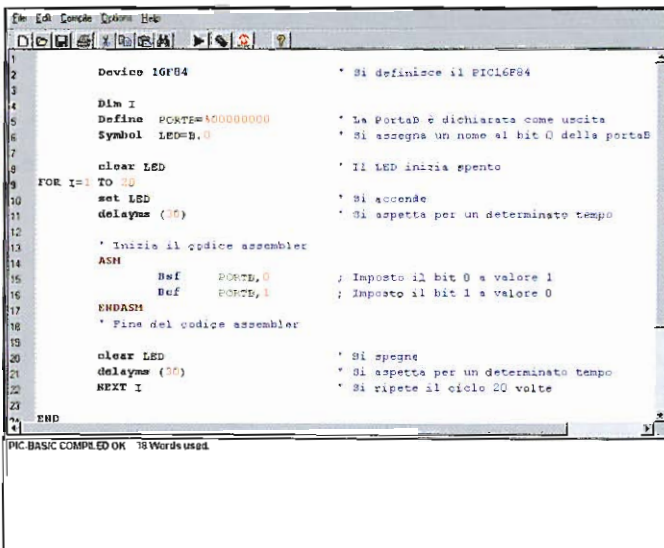
Eseguite una prima sostituzione, poi tornate dall'inizio e sostituite una seconda volta se è necessario, arriverà il momento in cui tutte le istruzioni contenenti la @ spariranno. A questo punto il risultato sarà l'assembler puro.



```
Device 16F84
Dim I
Define PORTB=10000000
Symbol LED=B,0
clear LED
FOR I=1 TO 20
set LED
delayms (50)
' Inizio il codice assembler
ASH
Bsf PORTB,0
Bcf PORTB,1
ENDASH
' Fine del codice assembler
clear LED
delayms (50)
NEXT I
```

Errors[123] C:\PROGRAM-FILE-1\PIC\PROGRAM-1\F36_GASM 26: Signal argument (expected single character)
Errors[122] C:\PROGRAM-FILE-1\PIC\PROGRAM-1\F36_GASM 3s: Missing operator
Errors[124] C:\PROGRAM-FILE-1\PIC\PROGRAM-1\F36_GASM 37: Bopul argument (expected single character)
Errors[125] C:\PROGRAM-FILE-1\PIC\PROGRAM-1\F36_GASM 37: Missing operator

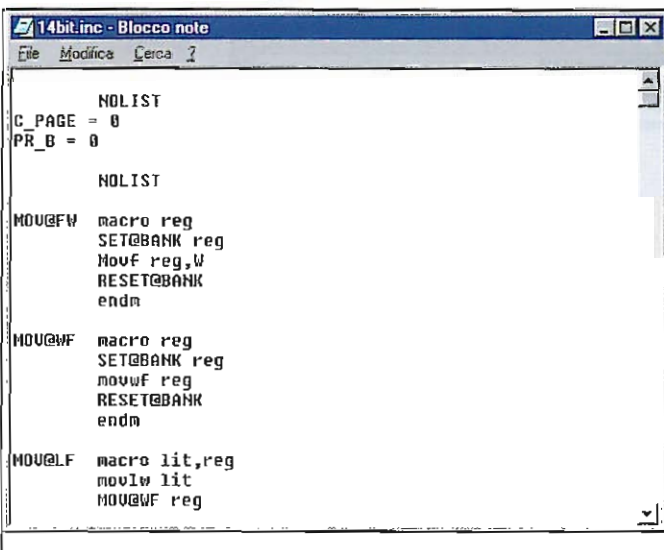
Esercizio che genera errori alla compilazione.



```
Device 16F84
Dim I
Define PORTB=10000000
Symbol LED=B,0
clear LED
FOR I=1 TO 20
set LED
delayms (50)
' Inizio il codice assembler
ASH
Bsf PORTB,0 ; Imposto il bit 0 a valore 1
Bcf PORTB,1 ; Imposto il bit 1 a valore 0
ENDASH
' Fine del codice assembler
clear LED
delayms (50)
NEXT I
END
```

PIC BASIC COMPILED OK 18 Words used.

Esercizio con commenti "comprensibili" dall'assembler.



```
14bit.inc - Blocco note
File Modifica Cerca ?
NOLIST
C_PAGE = 0
PR_B = 0
NOLIST
MOV@FW macro reg
SET@BANK reg
Mouf reg,w
RESET@BANK
endm
MOV@WF macro reg
SET@BANK reg
mouwf reg
RESET@BANK
endm
MOV@LF macro lit,reg
movlw lit
MOV@WF reg
```

Programma che contiene macro per un passo intermedio prima dell'assembler.