

## Strutture di controllo

Nel LetPicBasic avevamo a disposizione quattro strutture di controllo per costruire i nostri programmi, ora nel LetPicBasicPlus questo numero è stato ampliato in modo significativo, per facilitare la programmazione. Alcune sono uguali, o simili, in entrambi i programmi, altre però aggiungono funzionalità che in precedenza dovevamo costruire noi stessi.

### La struttura condizionale IF...THEN...ELSE...ENDIF

Nel LetPicBasic non esisteva alcuna istruzione che ci permettesse di realizzare strutture condizionali a due rami, infatti per far funzionare una struttura in questo modo dovevamo fare un piccolo "trucco". Con la struttura IF...THEN...ELSE...ENDIF questo si può ottenere facilmente, dato che abbiamo possibili formati.

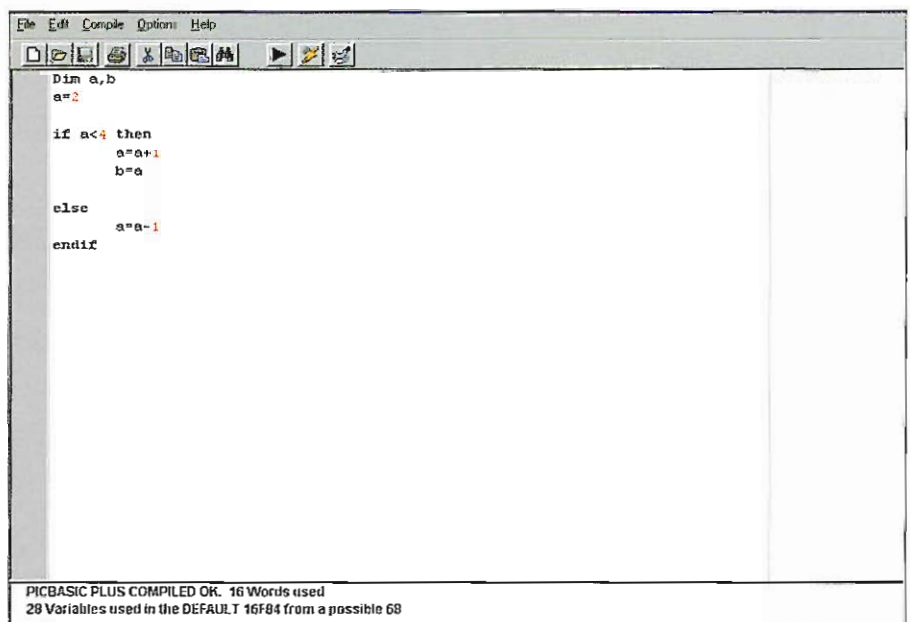
Con il primo formato, se si compie la condizione vengono eseguite le istruzioni poste di seguito sulla stessa linea. Se non si compie, queste istruzioni non vengono eseguite.

La separazione fra le istruzioni è realizzata con i due punti (:).

La terza versione, come possiamo vedere chiaramente dalla tabella in alto, è quella che abbraccia tutte le possibilità, dato che è possibile lasciare

```
IF condizione THEN istruzione:{istruzione}
IF condizione THEN istruzione:{istruzione}:ELSE istruzione: {istruzione}
IF condizione THEN
    istruzione
    {istruzione}
ELSE
    istruzione
    {istruzione}
ENDIF
```

Sintassi della struttura condizionale IF...THEN...ELSE...ENDIF.



```
File Edit Compile Options Help
[Icons]
Dim a,b
a=2

if a<4 then
a=a+1
b=a

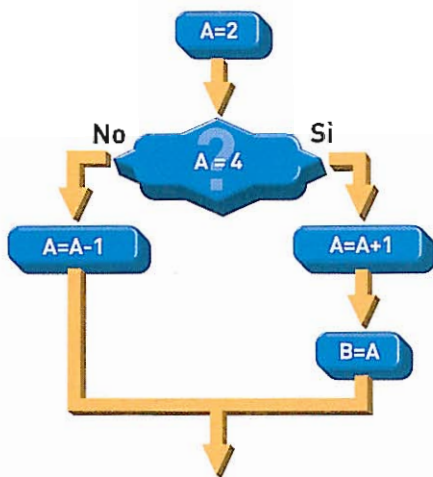
else
a=a-1
endif

PICBASIC PLUS COMPILED OK. 16 Words used
28 Variables used in the DEFAULT 16F84 from a possible 68
```

Esempio di utilizzo della struttura IF...THEN...ELSE...ENDIF.

parti vuote. Inoltre, per noi, è obbligatorio utilizzare quest'ultima modalità, infatti la versione Lite non ci permette di scrivere più di una istruzione sulla stessa linea. Nell'esempio della figura, dal

momento che la variabile "a" è minore di 4 vengono eseguite le due istruzioni poste dopo "then". Nel caso che "a" assuma un valore maggiore o uguale a 4 verrà eseguita l'altra istruzione. In ogni caso, verrà seguita una



Organigramma di esecuzione dell'esempio proposto.

sequenza sino a "endif". La forma che questo esempio adotterebbe in un organigramma è riportata nella figura in alto.

Anche se questo semplice esempio serve solamente per comprendere il funzionamento della struttura, non è normale che si assegni un valore a una variabile e dopo si esegua un test sul suo valore, dato che se ne conosce già il risultato, quindi il ramo delle istruzioni che verrà eseguito. Le variabili di solito contengono valori che vengono modificati lungo il corso del programma, perché subiscono le influenze degli elementi d'ingresso; in questo caso, quando si realizza la comparazione il risultato non dipende da noi.

## Le strutture **BRANCH** e **BRANCHL**

Quando si vuole comparare la stessa variabile con una serie di valori, ed eseguire azioni

```

IF condizione1 THEN
  istruzione
  {istruzione}
ELSE
  IF condizione2 THEN
    istruzione
    {istruzione}
  ELSE
    IF condizione3 THEN
      istruzione
      {istruzione}
    ELSE
      istruzione
      {istruzione}
    ENDIF
  ENDIF
ENDIF
ENDIF
  
```

Sintassi della struttura condizionale a rami multipli formata a partire da strutture IF...THEN.

```
BRANCH var, [etic1 {, ...eticn}]
```

```
BRANCHL var, [etic1 {, ...eticn}]
```

Sintassi della struttura **BRANCH** e **BRANCHL**.

differenti a seconda del valore stesso, se non si dispone dell'istruzione adeguata bisogna concatenare diverse istruzioni IF...THEN...ELSE...ENDIF. Nel LetPicBasicPlus abbiamo due nuove istruzioni: **BRANCH** e **BRANCHL**. Entrambe sono simili, perché saltano a un indirizzo segnato da un nome di etichetta a seconda del valore di una variabile.

La differenza consiste nel microcontroller con cui vengono

utilizzate. Se la memoria delle istruzioni è di una sola pagina, come succede con il PIC16F84, l'istruzione da utilizzare è la **BRANCH**. La **BRANCHL** si utilizza quando la memoria delle istruzioni del PIC è maggiore di una pagina e si sospetta che il salto possa cambiare pagina. Utilizzare un'istruzione **BRANCHL** quando è sufficiente una **BRANCH**, fa sì che la quantità di codice generato sia maggiore. Il primo salto si esegue quando la variabile vale 0, il secondo quando vale 1, ecc.; in questo modo, nell'esempio delle figure, quando la variabile assume il valore 2 salterà all'etichetta "lab3", eseguendo le istruzioni che si trovano di seguito.

Notate che come ultima istruzione di ogni indirizzo di salto, è stata posta un'istruzione "goto", in caso contrario si continuerebbe in sequenza fino all'etichetta successiva. La differenza fra queste due figure è il numero d'istruzioni macchina che generano, come si può vedere nell'informazione di compilazione di ognuna di esse: 37 rispetto a 25 istruzioni.

## Il salto incondizionato **GOTO**

Su questa istruzione non c'è nulla che non sia già stato detto. Nel LetPicBasicPlus segue la struttura che già conosciamo, e la dobbiamo usare con lo stesso rispetto e attenzione se vogliamo che sia un aiuto e che non rompa la programmazione strutturata.