

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

branch a, [lab1, lab2, lab3]
lab1:
  a=a+1
  goto fine
lab2:
  b=1
  goto fine
lab3:
  b=a
  goto fine

fine:
```

PICBASIC PLUS COMPILED OK. 25 Words used
28 Variables used in the DEFAULT 16F84 from a possible 68

Esempio di utilizzo della struttura **BRANCH**.

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

branchl a, [lab1, lab2, lab3]
lab1:
  a=a+1
  goto fine
lab2:
  b=1
  goto fine
lab3:
  b=a
  goto fine

fine:
```

PICBASIC PLUS COMPILED OK. 37 Words used
28 Variables used in the DEFAULT 16F84 from a possible 68

Esempio di utilizzo della struttura **BRANCHL**.

GOSUB Etichetta

```
...
Etichetta:
...
RETURN
```

CALL Etichetta

```
...
ASM
Etichetta
...
RETURN
ENDASM
```

Sintassi dell'esecuzione di subroutines.

Le subroutines

Abbiamo a disposizione due tipi di istruzioni per realizzare delle subroutines: GOSUB...RETURN e CALL...RETURN. Come possiamo vedere, la sintassi di entrambe le istruzioni è simile.

La differenza sta nel fatto che la seconda si utilizza se il codice della subroutine deve essere scritto in assembler. Nel caso in cui vengano annidate delle subroutines una all'interno dell'altra, bisogna ricordare che i PIC accettano solo 8 livelli di annidamento, e le librerie proprie del LetPicBasicPlus ne possono utilizzare sino a 4.

Le strutture di controllo ripetitive

Possiamo contare su tre tipi di strutture ripetitive, che sono quelle classiche nella maggioranza dei linguaggi di programmazione.

```

File Edit Compile Options Help
[Icons]
Dim a           ; Si dichiara la variabile "a"
GOTO start      ; Si salta alla posizione etichettata come "start"
ASR            ; Subroutine in assembler
etc1
  movlw 3
  return
ENDASH

etc2:           ; Subroutine in basic
  a=a+1
  return

start:         ; programma principale che chiama le due subroutines
  CALL etc1
  GOSUB etc2
    
```

PICBASIC PLUS COMPILED OK. 11 Words used
27 Variables used in the DEFAULT 16FB4 from a possible 68

Esempio d'esecuzione di subroutines.

```

File Edit Compile Options Help
[Icons]
Dim var, a
a=1
FOR var=10 TO 0 STEP -2 : ciclo FOR
  a=a+1
NEXT
a=1
var=10
WHILE var>=0 : ciclo WHILE
  a=a+1
  var=var-2
WEND
a=1
var=10
REPEAT
  a=a+1 : ciclo REPEAT
  var=var-2
UNTIL var<0
    
```

PICBASIC PLUS COMPILED OK. 43 Words used
28 Variables used in the DEFAULT 16FB4 from a possible 68

Lo stesso esempio con le tre strutture ripetitive.

FOR variabile=valore_iniziale TO
valore_finale [STEP passo_salto]

Istruzioni

NEXT

WHILE condizione

Istruzioni

WEND

REPEAT

Istruzioni

UNTIL condizione

Sintassi delle strutture ripetitive.

Nei cicli FOR...NEXT l'insieme di istruzioni verrà eseguito x volte, dove x dipende dai valori iniziali e finali che assumono le variabili e dall'incremento o dal decremento che subisce questa variabile con ogni nuova interazione. Il ciclo può essere ascendente o discendente. Con la struttura WHILE...WEND si ripeterà una sequenza di istruzioni mentre si sta compiendo una condizione e si continuerà in sequenza quando la condizione cessa.

In ultimo, la struttura REPEAT...UNTIL funziona in modo contrario alla precedente, dato che ripete una sequenza di istruzioni sino a che la condizione non si compie. La differenza fondamentale fra queste due strutture è che REPEAT...UNTIL viene eseguita almeno una volta e successivamente si verifica la condizione, mentre WHILE...WEND può anche non eseguire il blocco d'istruzioni se la condizione risulta falsa dall'inizio.