

Struttura di controllo

Nel LetPicBasic possiamo trovare quattro strutture di controllo: la condizionale IF...THEN, il salto incondizionato GOTO, la chiamata a subroutine GOSUB...RETURN e il ciclo FOR...NEXT. In questo paragrafo impareremo il loro funzionamento e vedremo come combinarli per creare espressioni più complesse.

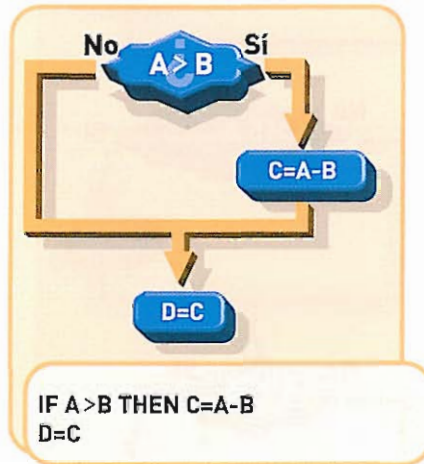
La struttura condizionale IF...THEN

Questa struttura segue la sintassi riportata nella figura, dove "comparazione" è un'espressione composta da variabili, numeri e comparazioni, ed "espressione" è l'istruzione che si esegue se la comparazione si compie.

```
IF {condizione} THEN {espressione}
```

Sintassi della struttura condizionale IF...THEN.

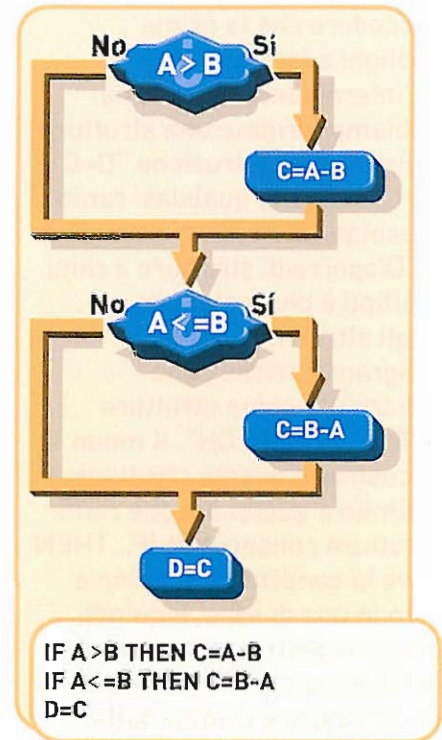
Se la comparazione non si compie, l'espressione è ignorata. Bisogna conoscere come si forma una struttura IF...THEN, dato che dopo l'istruzione da eseguire, si prosegue in sequenza. Nell'esempio della figura, se $A \rightarrow B$, a C verrà assegnato il risultato della sottrazione A-B. Successivamente il valore di C viene passato a D, senza tener conto della comparazione.



IF A > B THEN C=A-B
D=C

Esempio di utilizzo della struttura condizionale IF...THEN.

Queste strutture condizionali sono chiamate "ad un ramo", dato che hanno un solo possibile percorso da seguire. Normalmente se la comparazione si compie si



IF A > B THEN C=A-B
IF A <= B THEN C=B-A
D=C

Esempio di esecuzione di una struttura IF...THEN...ELSE a partire da strutture IF...THEN.

```
IF {condizione} THEN {espressione 1}
IF {no-condizione} THEN {espressione 2}
```

Sintassi della struttura IF...THEN...ELSE formata a partire da strutture IF...THEN.

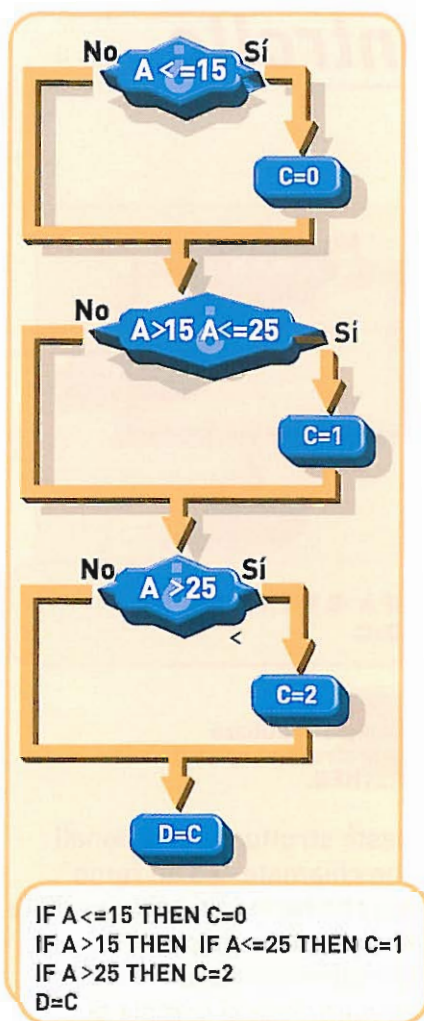
esegue una cosa, altrimenti un'altra. In altri linguaggi di programmazione, esiste per questo una struttura IF...THEN...ELSE. In questo caso non disponiamo di questa struttura, però la possiamo formare a

partire da due strutture IF...THEN. Notate che il trucco consiste nel fare due domande consecutive, ma con la comparazione al contrario. In questo modo solo una delle due

comparazioni diventa vera, quindi, si eseguirà una sola delle espressioni; in ogni caso per fare in modo che vengano eseguite entrambe le comparazioni, non deve succedere che la prima obblighi a fare un salto all'interno del programma. Abbiamo formato una struttura a due rami. L'istruzione "D=C" verrà eseguita qualsiasi ramo si scelga.

Disporre di strutture a rami multipli è piuttosto comune, negli altri linguaggi di programmazione sono conosciute come strutture "CASE" o "SWITCH". Il modo di costruire queste strutture è simile a quello dei due rami: strutture consecutive IF...THEN dove la condizione si compie solo in una di esse, e quindi, solo una delle espressioni verrà eseguita, anche se per poter passare tramite tutte le condizioni non bisogna inserire alcuna istruzione di salto "GOTO". Nell'ultimo esempio vogliamo introdurre anche le strutture IF...THEN annidate visto che, nel caso in cui ci sia necessità che si compiano due condizioni alla volta per realizzare un'azione, nel LetPicBasic non si può utilizzare l'operazione AND per operare con dati che non siano di tipo bit.

Nel caso in cui non si compia una delle condizioni non si esegue l'istruzione.



Esempio di esecuzione di una struttura a rami multipli formata a partire da strutture IF...THEN.

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

Il salto incondizionato GOTO

"GOTO" è un'istruzione tabù in molti linguaggi di programmazione, però avremo bisogno di utilizzarla quando lavoreremo in assembler quindi è meglio impararne il corretto utilizzo. La sua sintassi è semplice, e permette il salto a qualsiasi parte del programma, secondo quanto indicato dalla "etichetta", senza restrizioni. Questa mancanza di restrizioni la rende pericolosa, dato che un suo utilizzo improprio può rompere la struttura di un programma. Così, ad esempio, non si può utilizzare un'istruzione "GOTO" per saltare ad un punto intermedio di una subroutine a causa della confusione che questo può creare. Anche se la programmazione strutturata raccomanda di non utilizzarla, la mancanza di altre istruzioni nei linguaggi che utilizzeremo in futuro, la rende indispensabile.

GOTO {etichetta}

Sintassi dell'istruzione GOTO.

Se la uniamo alla struttura IF...THEN, aggiungiamo a quest'ultima una grande potenza, poiché ci permette non solo di eseguire un'istruzione o

```

IF {condizione 1} THEN {espressione 1}
IF {no-condizione 1 e condizione 2} THEN {espressione 2}
IF {no-condizione 1 e no-condizione 2 e condizione 3} THEN {espressione 3}
  
```