

Istruzioni di gestione delle tabelle dei dati

Il concetto visto nel paragrafo precedente è valido per qualsiasi linguaggio di programmazione, anche se i nomi possono variare fra "indice", "puntatore" o "tabella", "array", ecc. Il modo di lavorare con le tabelle, quindi, è simile in qualsiasi linguaggio anche se ognuno di essi dispone di istruzioni specifiche. In LetPicBasic le istruzioni sono "restore" e "read". Con "restore" si assegna all'indice un valore per fare in modo che punti alla posizione desiderata, tenendo conto che si inizia a numerare da 0. Non mettere nessun parametro dopo l'istruzione "restore" è come mettere uno 0, e al posto di un numero si può porre il nome di una variabile definita dall'utente. Con l'istruzione "read" si introduce il valore della posizione puntata dall'indice nella variabile data come parametro di "read". Prima di questo dovremo utilizzare l'istruzione "restore" per posizionare l'indice correttamente. Se le letture dei dati devono essere fatte in modo consecutivo, non serve utilizzare nuovamente l'istruzione "restore", dato che dopo ogni "read" l'indice si incrementa automaticamente di 1, puntando così alla posizione successiva. Il programma di esempio della figura, dopo aver definito la tabella posiziona l'indice all'inizio della stessa, mediante l'istruzione "restore". Non è necessario nessun parametro. Su "a" arriverà il valore di questa posizione mediante l'istruzione "read". Si otterrà così il valore "2". Se a questo punto si esegue

```

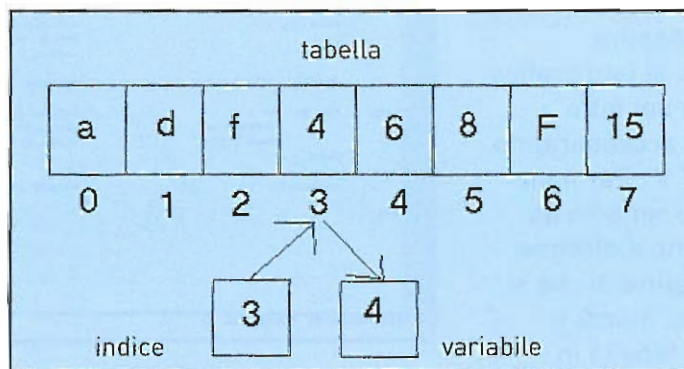
1 Device 16F84          * Si definisce il PIC16F84
2
3 dim a,b              * Definizione della variabile che conterrà
4                     * i dati e l'altra variabile
5
6 data 2,3,4,"red",00  * Si definisce una tabella di dati variabili
7
8 restore             * Si posiziona l'indice all'inizio della
9                     * tabella
10 read a              * Si legge la posizione puntata dall'indice
11
12 restore 3           * Si punta l'indice alla posizione 3
13 read a              * che punta al valore "c"
14
15 b="E"
16 restore b           * L'indice punta a "E"
17 read a
18
19 end
    
```

Programma esempio per la gestione della tabella dei dati.

```

1 Device 16F84          * Si definisce il PIC16F84
2
3 dim a,b,c,d,e        * Definizione delle variabili che conterranno
4                     * i dati
5
6 data 2,3,4,"red",00  * Si definisce una tabella di dati variabili
7
8 restore             * Si posiziona l'indice all'inizio della
9                     * tabella
10 read a              * Si legge la posizione puntata dall'indice
11 read b              * Seconda posizione
12 read c              * Valore "4"
13 read d              * "c"
14 read e              * "e"
15
16 end
    
```

Ogni valore può essere contenuto in una tabella, se questo ci interessa.



Schema del concetto di lavoro con le tabelle dei dati.

un altro "read" il valore che raccoglieremo sarà il "3", grazie all'auto incremento dell'indice di cui abbiamo parlato prima. I due esempi successivi posizionano il puntatore a un altro indirizzo e raccolgono il valore. La variabile "a" cambia così il valore ogni volta, anche se avremmo potuto realizzare le operazioni su diverse variabili, se ne

avessimo avuto bisogno.

Lavorando con queste tabelle bisogna fare attenzione a non confondersi e ad assegnare all'indice un valore non valido, per esempio un valore superiore all'ultima posizione della tabella; il compilatore non ci avviserà dell'errore, però eseguendo il programma le conseguenze saranno imprevedibili.

Tablelle per la visualizzazione di messaggi

Dopo aver visto il lavoro con le tabelle torniamo al problema che ne ha reso necessario l'utilizzo.

Volevamo un programma che mandasse dei messaggi su un display a 7 segmenti in modo continuo. Sappiamo che le tabelle contengono i dati in decimale, quindi le dovremo trasformare in questo codice, però i display a 7 segmenti hanno un codice proprio e la prima cosa da fare è tradurre un codice in un altro.

Per fare questo disegneremo uno schema come quello della figura, dove ogni lettera corrisponde a un segmento e "dp" è il punto decimale; dovremo tener conto, inoltre, che un display a 7 segmenti può essere ad anodo oppure a catodo comune, al lato pratico questo si traduce nel fatto che i segmenti si accenderanno mandando "0" o "1". Per ogni lettera, numero o simbolo da visualizzare faremo il disegno sul display dei segmenti che si devono illuminare, quindi li segnereemo sulla tabella in forma di "1" per il display a catodo comune e come "0" per il tipo ad anodo comune. Il valore risultante lo trasformeremo in decimale, e questo sarà il dato da introdurre nella tabella. Il passaggio da binario a decimale si realizza con la somma delle potenze di 2 per quelle posizioni che hanno valore 1.

Il programma che volevamo

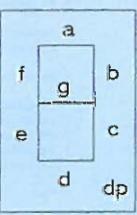
```

1 Device 16F04          ' Si definisce il PIC16F04
2
3 dim a                ' Definizione della variabile che conterrà
4                    ' i dati
5
6 data 2,3,4,"redP",00 ' Si definisce una tabella di dati variabili
7
8
9 restore 20           ' Si posiziona l'indice fuori dalla
10                    ' tabella
11 read a              ' Si legge la posizione puntata
12                    ' dall'indice 20
13
14 end
    
```

Il compilatore non ci avvisa di errori se noi usciamo dalla tabella.

Valori da mandare	dp g f e d c b a							Valore decimale
	0	0	0	1	1	1	1	
A	0	1	1	1	0	1	1	119

128+64+32+16+8+4+2+1



Catodo comune

Trasformazione dei valori da 7 segmenti a decimale.

```

1 Device 16F04          ' Si definisce il PIC16F04
2
3 Dim x                ' Variabile per l'indice
4
5 Data 63,119          ' Definizione della tabella dei dati
6                    ' con i valori "0" e "A" passati da
7                    ' codice a 7 segmenti in decimale
8
9 Define PORTB=00000000 ' Imposto la PortaB come uscita
10
11
12 Loop: For x=0 to 1   ' Estensione della tabella
13     restore x        ' Posizionamento dell'indice
14     read PORTB       ' Porta il valore sulla porta B
15
16     goto Loop        ' Eseguo il ciclo all'infinito
17
18 END
19
20
    
```

Visualizzazione dei dati in un display di 7 segmenti.

è riportato nella figura.

Dopo aver definito la tabella e dichiarato la porta B come uscita, che sarà la porta da collegare al display a 7 segmenti, il programma entra in un ciclo che percorre tutta la tabella dei dati, leggendo i valori, e portandoli sul display a 7 segmenti. Con l'utilizzo della variabile "X" per posizionare l'indice nella tabella,

noi ci assicuriamo che questo non prenda alcun valore errato.

Al termine del ciclo, un salto incondizionato lo rimanda nuovamente all'inizio dando così forma a un ciclo infinito. Introducete ora i vostri valori da visualizzare. Notate che per quanti valori vengano introdotti, il numero di istruzioni non cambia.