

**Benvenuti al
corso di
C.n.c.
Help on-line**

Corso di controllo numerico (Cnc iso)

A cura di Ferrarese Adolfo.

Prefazione:

Il corso si articola in **8** sezioni: **7 lezioni più un'ampia sezione tabelle** di uso comune nella meccanica

- 1- Programmazione elementare di base (lezione 1).
- 2- Programmazione di macchine fresatrici e centri di lavoro a CNC (lezioni 2 , 3 , 4).
- 3- Programmazione di torni a CNC (Lezioni 5 e 6).
- 4- Programmazione avanzata (parametrica con espressioni IF - GOTO) (lezione 7).**
- 5- Tabelle di uso comune nella meccanica (Velocità utensili, filettature ecc..) (lezione 8).

Lezione 7

Programmazione avanzata.

Durante lo svolgimento del nostro corso abbiamo imparato a stilare un listato con parecchi comandi e siamo oggi in grado di far eseguire alla macchina lavorazioni anche complesse. Il controllo numerico possiede però altre potenti funzioni, in grado di far risparmiare righe di codice da implementare (con l'aggiunta di iterazioni e loop) e allo stesso tempo di aggiungere versatilità ai programmi (macro utente, ovvero cicli personali per lavorazioni particolari costruiti con programmazione parametrica ed espressioni condizionate).

Nel trattare questa sezione del corso presumiamo che voi siate a conoscenza di un po' di matematica a livello di scuola media inferiore (espressioni matematiche e funzioni trigonometriche).

Descriveremo alcuni elementi di informatica di base, chiarendo il concetto delle variabili numeriche e delle espressioni condizionate ed incondizionate, parleremo di come caricare ed implementare i parametri (anche con complesse espressioni matematiche), quindi impareremo ad usare tutte queste funzioni insieme.

Una cosa che non vi possiamo insegnare e che è necessaria ad un buon programmatore è la capacità di analizzare i problemi da risolvere per lavorare un dato pezzo ed elaborare il programma prima nella mente e poi sul computer. Questa è una dote che dovete sviluppare con la pratica.

Informatica di base: le variabili.

Le variabili sono dei contenitori in grado di ricevere valori numerici o alfanumerici (stringhe di testo; non usate nei cnc) e che possono essere in qualsiasi momento aggiornate con un nuovo valore. Esse hanno bisogno di un nome univoco per essere gestite dal programmatore in fase di implementazione e dal computer in fase di elaborazione.

Per meglio capire immaginiamo di avere un'enorme cassetiera con centinaia di cassette: ognuno di questi può contenere qualcosa.

Per dire ad un nostro amico di andare a prendere un oggetto in un determinato cassetto dobbiamo indicargli dove questo cassetto si trova.

Se dunque noi diamo un nome ad ogni cassetto basterà dirgli di aprire il cassetto con quel nome specifico e subito il nostro amico prenderà l'oggetto giusto.

Usiamo dunque il nome più semplice e corto possibile per i nostri cassette: i numeri.

Il primo cassetto si chiamerà 1; il secondo 2..... il duecentesimo 200.

Se nel cassetto 32 mettiamo una chiave sappiamo che in quel cassetto ci sarà una chiave fino a quando non cambieremo quel contenuto aggiungendo o sottraendo chiavi.

Dunque il nome variabile deriva dal fatto che il contenuto può appunto variare. Il computer non ha cassetiere ma celle di memoria, non contiene chiavi ma valori numerici ma il funzionamento della gestione delle variabili è uguale.

Questa funzione è molto utile e potente in quanto ci permette di inserire dati e riutilizzarli ogni volta che ne abbiamo bisogno senza ricordarli e riscriverli tutte le volte, inoltre le variabili possono essere caricate (caricare = inserire valori) direttamente (variabile 1 = 120,567) oppure attraverso operazioni matematiche (variabile 1 = $467,569 + 456,9$).

I parametri: premessa.

I parametri non sono altro che variabili usate all'interno di un controllo numerico; essi funzionano esattamente come nella descrizione soprastante.

Nel cnc anche i parametri vanno indicati come tutti i comandi ovvero con un codice ed un indirizzo numerico che insieme formano il nome del parametro stesso.

Come sempre il codice cambia in funzione del costruttore del controllo numerico: Siemens usa la 'R' Heidenhain usa la 'Q' Mitsubishi usa il cancelletto '#'.

Non importa quale simbolo usi la macchina con cui avete a che fare, l'importante è comprendere il concetto di

base che è uguale per tutti i cnc.

In questa sezione del corso tratterò la programmazione avanzata descrivendo, per semplicità, solo i controlli numerici **Mitshubishi serie Meldas 500**, cogliete dunque i concetti e riferitevi al manuale della vostra macchina per l'esatta sintassi.

I parametri: uso.

Nelle macchine utensili cnc vi sono due modi di inserire i valori numerici.

Assegnazione diretta: è il metodo che abbiamo imparato durante il corso; **G0 X100 Z-20** sono assegnazioni dirette del valore numerico.

Ma esiste la possibilità di assegnare, dopo il codice, un parametro in luogo del valore numerico. E' concesso l'uso dei parametri con quasi tutti i codici ad eccezione di **O**, **N** e **/**.

Assegnazione indiretta: è dunque il metodo di inserimento dei valori numerici tramite l'uso di un parametro; **G#1 X#2 Z#3** sono assegnazioni indirette del valore numerico.

Per utilizzare i parametri occorre prima di tutto assegnarli un valore (caricarli) quindi richiamarli all'interno della programmazione.

Non è possibile assegnare il valore del parametro nella stessa riga in cui esso viene utilizzato poiché occorre dare il tempo al calcolatore di registrare il valore del parametro prima di poterlo usare:

#2=100 X#2 (errato);

X#2=100 (errato);

**#2=100;
X#2 (corretto);**

Inoltre è possibile caricare i parametri attraverso operazioni matematiche; se esse sono complesse per esempio espressioni, si devono usare coppie di parentesi (fino ad un massimo di 5 coppie) per evitare operazioni ambigue.

Il calcolatore segue la priorità consueta nello svolgimento delle operazioni quindi la seguente espressione potrebbe non dare il risultato sperato:

5 + 8 / 2 = 9

Questo perché viene risolta prima la divisione ($8 / 2 = 4$) e quindi l'addizione ($5 + 4 = 9$), se si intendeva dividere per 2 il risultato dell'addizione occorre mettere i numeri da sommare tra parentesi (nel nostro cnc esse sono quadre):

[5 + 8] / 2 = 6,5.

Attenzione dunque a queste ambiguità.

Nell'esempio precedente (**G0 X100 Z-20**) dovremmo procedere come segue:

N10 #1=0 #2=100 #3=-20 (Carico i parametri)
N20 G#1 X#2 Z#3 (assegnazione indiretta degli indirizzi)

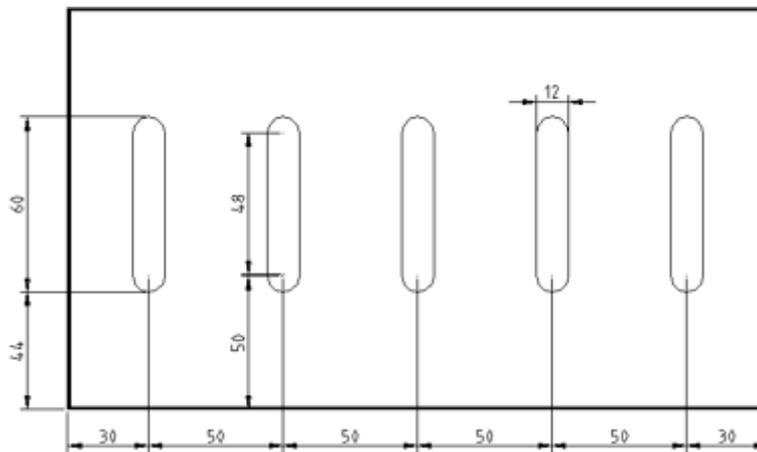
NB: alcuni cnc interpongono un = tra codice e parametro (Siemens usa **G= R1 X= R2 Z= R3**).

L'uso dei parametri è tutto qui! Una volta capito il funzionamento abbiamo a disposizione uno strumento molto potente per la nostra programmazione.

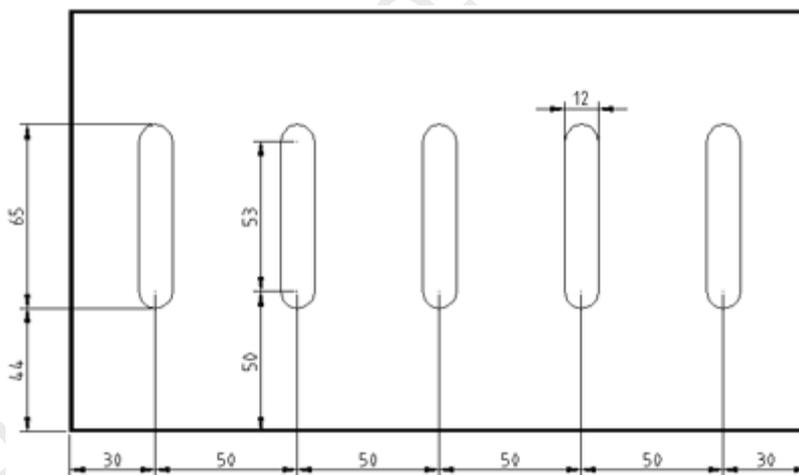
Certo qualcuno potrà sollevare l'eccezione che assegnando direttamente il valore si sarebbe risparmiato una riga di codice. In effetti questo era solo un semplice esempio fatto per capire come utilizzare la parametrica ma vedremo più avanti come essa è efficace e risolutiva di problemi altrimenti insormontabili.

Facciamo un semplice esempio utilizzando prima la programmazione tradizionale poi quella con l'uso dei parametri e quindi quella con l'uso della sottoprogrammazione abbinata ai parametri per avere il massimo di flessibilità.

Si supponga di dover eseguire una serie di fori asolati non interpolati su un due pezzi che sono simili ma non uguali come nei disegni sottostanti:



Disegno 1



Disegno 2

Si noti che i due pezzi differiscono soltanto nell'altezza del foro asolato (il primo 60 mm, il secondo 65 mm) e quindi dobbiamo fare un programma unico per tutti e due i disegni.

Con il metodo convenzionale dovremmo programmare il primo disegno, definendo la lavorazione dell'asola una ad una, quindi terminata la lavorazione avremmo dovuto cambiare tutte le quote Y dell'asola stessa con un dispendio di lavoro e anche con il rischio di sbagliare.

Vediamo la programmazione tradizionale dei pezzi dei disegni soprastanti; UTENSILE = FRESA DIAMETRO 12

Disegno 1

```

N10.....
G0 X30 Y50
G1 F50 Z-10
Y98
Z3
G0 X80 Y50
G1 F50 Z-10
Y98
Z3
G0 X130 Y50
G1 F50 Z-10
Y98
Z3
G0 X180 Y50
G1 F50 Z-10
Y98
Z3
G0 X230 Y50
G1 F50 Z-10
Y98
Z3
.....M30

```

Disegno 2

```

N10.....
G0 X30 Y50
G1 F50 Z-10
Y103
Z3
G0 X80 Y50
G1 F50 Z-10
Y103
Z3
G0 X130 Y50
G1 F50 Z-10
Y103
Z3
G0 X180 Y50
G1 F50 Z-10
Y103
Z3
G0 X230 Y50
G1 F50 Z-10
Y103
Z3
.....M30

```

In rosso i dati da cambiare per poter eseguire i nostri pezzi. Come vediamo ad ogni asola dobbiamo cambiare il valore di Y.

Con i parametri tutto risulta più semplice:

Disegno 1

N10.....

N15 #100=98 (definizione del parametro)

G0 X30 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 98 nella riga 15))

Z3

G0 X80 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 98 nella riga 15))

Z3

G0 X130 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 98 nella riga 15))

Z3

G0 X180 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 98 nella riga 15))

Z3

G0 X230 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 98 nella riga 15))

Z3

.....M30

Disegno 2

N10.....

N15 #100=103 (Unico dato da variare rispetto al programma del disegno 1)

G0 X30 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 103 nella riga 15))

Z3

G0 X80 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 103 nella riga 15))

Z3

G0 X130 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 103 nella riga 15))

Z3

G0 X180 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 103 nella riga 15))

Z3

G0 X230 Y50

G1 F50 Z-10

Y#100 (assegno ad Y il valore contenuto
nel parametro 100 (posto a 103 nella riga 15))

Z3

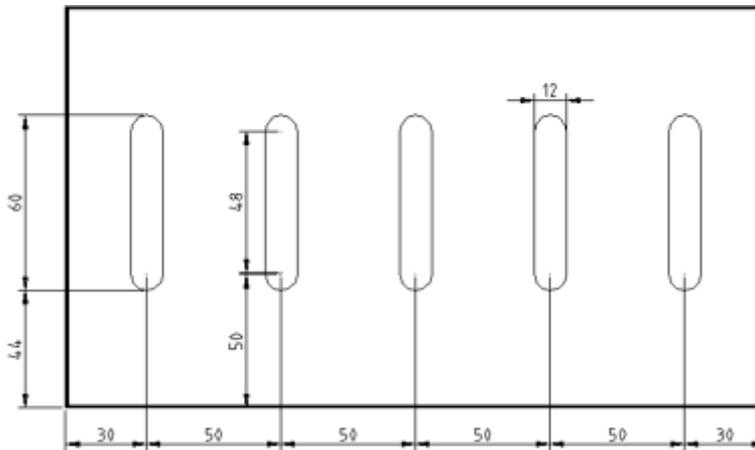
.....M30

Notiamo quanto l'uso dei parametri semplifichi la programmazione riducendo la possibilità di errore. Se essa è abbinata alla sottoprogrammazione essa diviene ancora più versatile con ulteriore risparmio di scrittura di codice come vedremo nella pagina successiva.

I parametri nella sottoprogrammazione.

Ricordando che il codice qui scritto è valido soltanto per i cnc Mitsubishi serie meldas 500 o simili, vediamo come l'uso della sottoprogrammazione abbinata alla parametrica porti a diminuire la quantità di codice da implementare aumentando la versatilità del codice stesso.

Programmiamo il pezzo 1 dell'esempio della pagina precedente:



Disegno 1

N10.....	
N20 #100=98	(quota di traslazione Y \ Inizio definizione dei parametri)
N22 #101=50	(distanza tra i fori asolati in X)
N25 #102=30	(primo foro asolato in X \ fine definizione dei parametri)
N30 M98 H50 L5	(richiamo sottoprogramma interno che ha inizio dalla riga 50 fino a M99)
M30	(Fine programma)
N50	(inizio sottoprogramma)
G0 X#102 Y50	(#102 è posto a 30 nella riga 25)
N60 G1 F50 Z-10	
N70 Y#100	(#100 è posto a 98 nella riga 20)
Z3	
N80 #102=#102+#101	(Incremento del parametro 102 per l'asola successiva)
M99	(fine sottoprogramma)

Questo è il programma per eseguire il pezzo di cui sopra; notiamo quanto poco codice è stato scritto e capiremo, dopo aver spiegato il programma, quanto versatile può esso diventare.

Alla riga 10 carichiamo l'utensile, definiamo i giri del mandrino ecc.

Alla riga 20 carichiamo i parametri che useremo nel programma, essi sono 3: il parametro 100 conserva la quota da raggiungere con l'asse Y nell'esecuzione dell'asola.

Il parametro 101 viene caricato con il valore di distanza tra i fori asolati che è per tutti di 50mm:

Infine il parametro 102 viene momentaneamente caricato con la distanza X della prima asola che è 30mm:

La riga 30 richiama il sottoprogramma di riga 50 all'interno del programma principale con il comando M98 H50. Il comando L5 richiama tale sottoprogramma per 5 volte consecutive prima di continuare con il programma principale alla riga successiva. Una volta eseguito il sottoprogramma per 5 volte. Il programma termina con M30

Il sottoprogramma è quello che esegue il lavoro ed ha inizio alla riga 50; abbiamo un posizionamento in rapido per l'asse X alla quota presente all'interno del parametro 102 che è 30mm e per l'asse Y a 50mm.

Alla riga 60 si scende a velocità lavoro (F50) a Z-10.

Nella riga 70 muoviamo l'asse Y fino alla quota contenuta nel parametro 100 che è stata posta a 98. Ma il cuore del programma è la riga 80; essa aggiorna il parametro 102 per l'asola successiva vediamo come;

Il parametro 102 ha in questo momento un valore di 30. Scrivere **#102=#102+#101** significa voler dire che si intende aggiungere al valore contenuto in quel momento dal parametro 102 il valore contenuto nel parametro 101.

parametro 102 = 30+
 parametro 101 = 50=
 80

Il parametro 102 ha quindi ora un valore pari ad 80.

Alla riga successiva il sottoprogramma termina con M99 ma il programma principale che ha per ora eseguito una sola delle cinque volte previste il sottoprogramma, rimanda in esecuzione il sottoprogramma a partire dalla riga 50.

E' chiaro che senza la riga 80 verrebbe ripetuta ancora la prima asola!

Ma, grazie alla riga 80 il parametro 102 ha assunto un valore pari ad 80 che è appunto la quota X della seconda asola.

Abbiamo dunque un posizionamento in rapido per l'asse X alla quota presente all'interno del parametro 102 che è ora di 80mm e per l'asse Y a 50mm e così via fino alla riga 80 che somma al valore del parametro 102 il valore del parametro 101 (80 + 50 = 130) predisponendo così la quota per la terza asola e così via fino all'esecuzione di tutte e cinque le asole.

Possiamo fare delle considerazioni su questo programma per dimostrare la versatilità che può esso assumere:

1- Cambiando semplicemente il valore del parametro 100, possiamo eseguire anche il disegno 2

2- Se ci capitasse un pezzo uguale ma la distanza tra le asole fosse diversa da 50 variando il valore del parametro 101 non avremo problemi, se fosse diversa la partenza della prima asola cambiamo il valore del parametro 102 e partiamo a lavorare.

3- Se la stessa ditta ordina pezzi uguali al disegno 1 ma più lunghi e quindi, a parità di dimensioni e di interrasse, occorresse fare diciamo quindici anziché cinque asole? Bel problema con la programmazione tradizionale ma con la sottoprogrammazione abbinata ai parametri basta cambiare L5, alla riga 30, con L15. Il gioco è fatto!!!

Abbiamo dunque visto quanti vantaggi offre questo modo di programmare e quanto potenti siano i parametri.

Espressioni di controllo condizionate ed incondizionate, introduzione.

Fino a questo punto abbiamo imparato che il programma da noi stilato viene eseguito dal computer a partire dalla prima riga e, se non trova interruzioni (M00) o non viene arrestato, fino alla fine del listato oppure fino al comando M30.

Ma non sempre è così, abbiamo visto che per esempio si può deviare l'esecuzione ad un sottoprogramma, vi è inoltre la possibilità di deviare il normale flusso di esecuzione, a righe d'istruzione superiori o sottostanti, ad un dato punto del programma attraverso espressioni condizionate od incondizionate.

Espressioni IF - GOTO

In determinate situazioni abbiamo la necessità di deviare il flusso di esecuzione del programma in una direzione diversa da quella che normalmente sarebbe percorsa dal programma.

Abbiamo due modi diversi per farlo:

1- Con un'espressione incondizionata.

E' la più semplice: si invia l'esecuzione del programma ad una riga di istruzione prestabilita in ogni caso

(incondizionata = senza alcuna condizione da valutare), ogni volta che il programma incontra tale istruzione. In genere tale istruzione viene usata vicino ad un'espressione condizionata.

2- Con un'espressione condizionata.

L'espressione condizionata devia il flusso di esecuzione verso un determinato numero di programma; in questo modo abbiamo la possibilità di ramificare il nostro programma verso due o più direzioni a seconda delle nostre necessità. L'uso di questa tecnica di programmazione è maggiormente impiegato per creare dei LOOP (collegare in circuito), in modo da far eseguire un gruppo di righe di istruzione un numero di volte necessarie allo svolgimento del lavoro da svolgere e quindi uscire per proseguire lo svolgimento del programma stesso. Per fare ciò occorre deviare il flusso di esecuzione verso un determinato numero di programma posto verso l'inizio dello stesso, finché una condizione da noi determinata viene soddisfatta, quindi si prosegue nella normale esecuzione.

Questi concetti appariranno più comprensibili con la seguente spiegazione e con i prossimi esempi.

Espressioni di controllo condizionate ed incondizionate, uso.

1- Espressione incondizionata.

Si inserisce semplicemente il comando **GOTO * ***

dove in luogo degli asterischi si indica la riga d'istruzione dove si desidera proseguire il programma esempio **GOTO 10** = vai alla riga 10. Questa istruzione può essere utilizzata da sola ad esempio per saltare una parte di programma che riguarda una lavorazione che solitamente facciamo eseguire ma che, per un particolare motivo, su alcuni pezzi non va fatta. Potremmo cancellare quella parte di programma, ma quando servirà ancora non l'avremmo più. Con l'istruzione **GOTO * *** risolviamo il problema: saltiamo la parte di programma che momentaneamente non serve e allo stesso tempo conserviamo la programmazione per quando servirà nuovamente, semplicemente cancellando quell'istruzione il programma sarà di nuovo interamente eseguito. Più spesso questa istruzione la troviamo in concomitanza di espressioni condizionate.

2- Espressione condizionata.

La condizione viene posta con la parola inglese **IF** (se).

Ad essa si associa un operatore di confronto.

Eccoli descritti con la loro simbologia ed il loro significato:

Simbolo convenzionale	Significato	Codice per il CNC
>	Maggiore di	GT
<	Minore di	LT
=	Uguale a	EQ
≥	Maggiore di o uguale a	GE
≤	Minore di o uguale a	LE
≠	Non uguale a (diverso da)	NE

Tra IF e l'operatore di confronto si inserisce un parametro che, di solito, funge da contatore. Dopo l'operatore di confronto si inserisce un parametro o un valore numerico che è il valore messo in confronto con il primo parametro.

Ecco come viene inserita nel cnc l'espressione condizionata:

```
N10 #100=1
N20.....
```

```
N80 IF [ #100 LT 10 ] GOTO 20
```

```
N90.....M30
```

L'espressione condizionata deve essere posta tra parentesi quadre; la condizione inserita al suo interno, nel caso della riga 80, pone questa domanda: "Il parametro 100 è inferiore a 10 ?" se la risposta è sì, la condizione risulta vera (è vero che #100 è minore di 10) e quindi la condizione è soddisfatta. Al contrario risulterà falsa e quindi insoddisfatta. Ricordate questo discorso quando sentirete parlare di condizioni vere e false, soddisfatte ed insoddisfatte poiché questo è il linguaggio dei programmatori.

La riga 80 comanda al cnc di tornare alla riga 20 (GOTO 20) fintanto che il parametro 100 rimane inferiore a 10

(#100 LT 10). Poiché alla riga 10 abbiamo posto a 1 il valore del parametro 100, esso è inferiore a dieci e quindi la condizione è soddisfatta (il parametro 100 è inferiore a 10) e quindi il programma, arrivato alla riga 80 salta (GOTO = vai a) alla riga 20.

Attenzione al rischio, come in questo caso, di creare loop infiniti.

In effetti nell'esempio precedente il parametro 100 resterà sempre di valore 1 e il programma non eseguirà mai la riga 90 in quanto l'espressione di riga 80 rimarrà sempre soddisfatta, se è di questo che avete bisogno il problema non si pone ma in genere non è mai così.

Risulta chiaro che per uscire da questo loop occorre che ad un certo punto il parametro 100 diventi uguale a 10! essere uguale a 10 infatti non soddisfa più la condizione che è vera soltanto finché il parametro 100 risulta inferiore.

Occorre dunque inserire all'interno del loop un comando che porti il parametro 100 al valore di 10 in questo modo:

```
N10 #100=1
```

```
N20.....
```

```
N30 #100 = #100 + 1
```

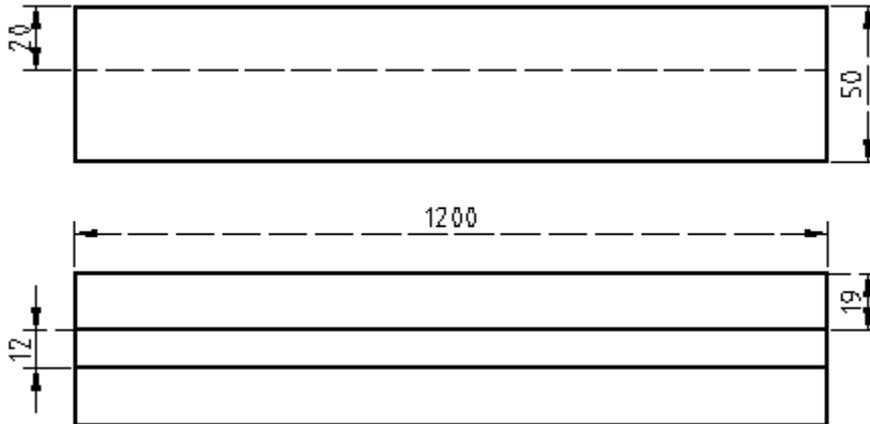
```
N80 IF [ #100 LT 10 ] GOTO 20
```

```
N90.....M30
```

La riga 30 è il classico contatore, esso ad ogni passaggio incrementa di 1 il parametro 100, come abbiamo già spiegato all'inizio di questa pagina (vedi), quindi quando il programma arriva alla riga 80 la prima volta troverà il parametro 100 con valore 2. La condizione è soddisfatta e quindi il programma salta alla riga 20. Quando torna alla riga 80 troverà il parametro 100 con valore 3, ripeterà la procedura appena descritta fino a quando il parametro 100 assumerà un valore di 10 rendendo falsa la condizione e quindi permettendo

l'uscita dal loop e l'esecuzione della riga 90.

Facciamo un esempio pratico, usando parametri ed espressioni condizionate ed incondizionate all'interno di un loop per l'esecuzione di una gola su piú passate:



Dobbiamo eseguire una cava di 12 mm di larghezza X 20 mm di profondità. Poiché utilizziamo una fresa diametro 12 in acciaio, dobbiamo eseguire passate con asportazione non superiore ai 4 mm. Il programma partirà da X-9 e arriverà ad X1209 con Z-4 quindi tornerà a X-9 con Z-8 e così via fino a Z-20.

```

N10 T1 M6 (fresa 12 mm)
G54 G90 G00 X-9 Y-25.....
....
#100=-9 #101=1209 (Punti X iniziale e finale)
#102= -4 (Incremento Z per passata)
#103=0 (Quota Z)
#104=30 (Velocità lavoro)
#105=-20 (Quota Z finale)
G0 Z3
N20 M98 H100
G0 Z200 M5
M30

N100G0 X#100 Y-25
Z3

N110 #103=#103+#102 (inizio loop)
G1 F#104 Z#103
X#101(Prima passata)

N120 IF [#103 EQ #105 ] GOTO 200

#103=#103+#102
F#104 Z#103
X#100 (Seconda passata)

GOTO110 (Fine loop)

N200 M99
  
```

Poiché dobbiamo programmare l'andata e il ritorno di Z, il loop deve interrompersi a metà e non alla fine in quanto al suo interno sono programmate due passate. Dobbiamo ripetere il loop tre volte ma se non uscissimo a metà del terzo passaggio eseguiremo sei incrementi Z da 4 mm ($6 \times 4 = 24$). Così facendo

invece ne possiamo eseguire solo cinque ($5 \times 4 = 20$) arrivando correttamente all'esatta quota Z.

Poniamo in questo caso l'espressione condizionata a metà loop e l'espressione incondizionata a fine loop. Questo è un ottimo esempio dell'uso in coppia dei due tipi di espressioni di controllo.

Oltre alle espressioni appena descritte, tutti i cnc danno la possibilità di usufruire di tutte le più comuni funzioni matematiche e trigonometriche, quali: radice quadrata, resto, esponente, seno, coseno, tangente etc.. Queste funzioni, insieme a quelle in questa lezione descritte, offrono uno strumento di programmazione potente il cui limite è dato solo dall'abilità del programmatore. L'esercizio e l'applicazione sono un ottimo metodo per affinare tale abilità.

Molto ci sarebbe da dire su queste tecniche di programmazione, tuttavia questo corso fornisce già un'ottima base per iniziare; esso fornisce nozioni essenziali, senza creare inutili confusioni (questo almeno è stato il mio intento), trattando gli argomenti di uso più comune.

Qui termina il corso di C.n.c. Help on-line. Nel prossimo ed ultimo capitolo pubblichiamo alcune tabelle di uso comune nella meccanica (12 pagine), vi consigliamo di visitare anche quest'ultima sezione, tali tabelle vi potrebbero tornare molto utili.

Questo corso è stato redatto in mesi di lavoro, oltre ad essere vietato dalla legge che tutela il copyright, copiare questo lavoro vuol dire offendere lo sforzo di chi, gratuitamente, si è impegnato per trasmettere un pizzico di conoscenza a quanti ne possono avere bisogno. Essere riuscito ad insegnare qualcosa a qualcuno di voi mi ripaga dello sforzo profuso nel redigere questo corso.

Per questo motivo fareste cosa gradita nell'esprimere la vostra opinione su questo corso a 865.180@email.it; complimenti o critiche che siano, servono per migliorare sempre di più questo servizio.

Per quanto riguarda molte delle illustrazioni presenti in questo testo si ringrazia l'istituto S. Zeno di Verona che le ha fornite.

Grazie a voi tutti per la dedizione con cui avete seguito il corso.

Ferrarese Adolfo.

[Termina qui la settima ed ultima lezione del corso di C.n.c. Help on-line.](#)

Arrivederci alla prossima sezione:

Tabelle di uso comune nella meccanica (Velocità utensili, filettature ecc..).