CAPITOLO 2
ESERCITAZIONI NUMERICHE DI PROGRAMMAZIONE

Questo capitolo è dedicato ad esercitazioni numeriche da eseguire con il P.C. per imparare la tecnica di programmazione in Qbasic mediante l'uso di tutte le funzioni riportate nei paragrafi dal 1.9.1 al 1.9.8. I paragrafi relativi alle esercitazioni sono intercalati da paragrafi inerenti a consigli e procedure di carattere operativo e di gestione programmi.
I programmi che andremo a compilare saranno di matematica generale senza riferimenti specifici a problematiche tecniche, argomento questo che sarà affrontato più avanti.
Gli esercizi permetteranno le manipolazioni delle funzioni elementari, singolarmente e nelle diverse strutture combinate quali funzioni miste, di più variabili, parametriche, funzioni di funzioni.

Salvo indicazioni diverse tutti gli argomenti che compariranno nelle funzioni trigonometriche saranno sempre espressi in radianti.

Ad eccezione del primo esercizio non si indicherà mai la pressione del tasto (invio) **che deve essere premuto dopo ogni digitazione** di istruzione o valore numerico.
E' utile rimarcare che le istruzioni possono essere digitate con lettere minuscole che, dopo il comando (invio), se digitate correttamente, vengono modificate automaticamente in lettere maiuscole dando così all'operatore la sicurezza di non aver commesso errori di sintassi.
Ad eccezione del primo esercizio non si specificherà più la pressione del tasto F5 indicando soltanto il simbolo F5 per un solo comando e 2F5 per due comandi in successione.
Ad eccezione del primo esercizio non si farà mai riferimento al paragrafo in cui sono riportate le corrispondenze tra funzioni in simbologia ordinaria e funzioni in simbologia Qbasic; sarà soltanto citato il numero d'ordine con il quale sono distinte.

2.1 Esercizio di programmazione n° 1 (funzione aritmetica)

Ci proponiamo di computare la funzione

$$Y = (x)^{1/2}$$

per tre valori della variabile indipendente: x = 7; x = 12; x = 3 .75
allo scopo, vista la corrispondenza simbolica 1) di paragrafo 1.9.1 che riportiamo:

Simbologia ordinaria	Simbologia Qbasic
1) (radice quadrata)	
$Y = (x)^{1/2}$	**Y = SQR (x)**

compiliamo il programma digitando le seguenti istruzioni:

```
CLS                (invio)
INPUT " x " ; x    (invio)
Y = SQR (x)        (invio)
PRINT "Y = " ; Y   (invio)
```

terminata la stesura del programma procediamo all'esecuzione:

(premere F5) e digitare il primo valore di x:

x ? 7 (invio)
Y = 2.645751
(premere due volte F5) e digitare il secondo valore di x :
x ? 12 (invio)
Y = 3.464102
(premere due volte F5) e digitare il terzo ed ultimo valore di x:
x ? 3.75 (invio)
Y = 1.936492
(premere F5) per ritornare alla schermata di programma

2.2 Esercizio di programmazione n° 2 (funzioni aritmetiche di due variabili)

Ci proponiamo di computare, contemporaneamente, le tre funzioni di due variabili

$Y = X1 \cdot X2$ $Y = X1 : X2$ $Y = (X1)^{x2}$

che hanno le corrispondenze simboliche in Qbasic rispettivamente nella 3), 4), 2), per le seguenti coppie di valori delle variabili indipendenti:

X1 = 3 X2 = 5
X1 = 2.5 X2 = .97

Per far ciò e necessario attribuire a ciascuna funzione un simbolo diverso in Y; Y1 per la prima, Y2 per la seconda, Y3 per la terza e compilare il programma come segue:

```
CLS

INPUT "X1" ; X1

INPUT "X2" ; X2

Y1 = X1 * X2

Y2 = X1 / X2

Y3 = X1 ^ X2

PRINT"Y1=" ; Y1

PRINT"Y2=" ; Y2

PRINT"Y3=" ; Y3
```

F5

X1 ? 3
X2 ? 5
Y1 = 15
Y2 = .6
Y3 = 243

2F5

X1 ? 2.5
X2 ? .97

```
Y1 = 2.425
Y2 = 2.57732
Y3 = 2.432214

                F5
```

2.3 Osservazioni in merito alla precisione di calcolo

Nei due esercizi svolti in precedenza abbiamo ottenuto risultati espressi con sei cifre decimali; se si desidera ottenere una precisione maggiore (doppia precisione) si deve posporre alla variabile dipendente il simbolo # come indicato: Y# .
Nel paragrafo seguente si applica questa nuova simbologia nel compilare uno dei programmi oggetto dell'esercizio n° 3.

2.4 Esercizio di programmazione n° 3 (funzione trigonometrica elementare)

Eseguiamo il calcolo della funzione Y = Sen x per il valore di x = .5 (radianti), utilizzando la corrispondenza simbolica 10) e scriviamo:

```
CLS

INPUT " x " ; x

Y = SIN ( x )

PRINT " Y = " ; Y

                F5
x ? .5
Y = .4794255
                F5
```

Il valore della variabile dipendente che abbiamo ottenuto è a singola precisione. Se si vuole sviluppare il calcolo in doppia precisione si deve ricompilare il programma in base a quanto indicato nel paragrafo 2.3:

```
CLS

INPUT"x" ; x

Y# = SIN ( x )

PRINT "Y" ; Y#

                F5

x ? .5
Y = .479425538604203

                F5
```

2.5 L'impiego delle costanti

In alcuni casi di compilazione dei programmi devono essere introdotte più costanti uguali; per semplificare questa procedura è sufficiente introdurre la costante una sola volta mediante l'uguaglianza con una lettera, a piacere, e richiamare poi tale lettera nell'ambito delle espressioni matematiche al posto della costante.

Se ad esempio la costante è π prima delle istruzioni che la riportano scriveremo, come fosse una istruzione, l'uguaglianza: p = 3.141592654

2.6 Esercizio di programmazione n° 4 (funzione trigonometrica combinata)

Prima di eseguire l'esercizio è opportuno osservare che con questo paragrafo si introducono le funzioni composte, funzioni che sono il risultato di diverse operazioni, aritmetiche ed analitiche. Questo tipo di funzioni sono utilizzate spesso nel testo, per tale ragione si deve spiegare, una volta per tutte, qual è il criterio che richiama le diverse corrispondenze simboliche.
Se la funzione da computare è ad esempio:

Y = Sen x Cos x + Tang x

troviamo nell'ordine: la funzione Sen x alla quale è associata la corrispondenza 10) **SIN (x)**, il prodotto tra Sen x e Cos x al quale è associata la corrispondenza 3) *, la funzione Cos x alla quale è associata la corrispondenza 11) **COS(x)**, la somma del prodotto (Sen x Cos x) con Tang x alla quale è associata la corrispondenza 5) +, ed infine la funzione Tang x alla quale è associata la corrispondenza 12) **TAN (x)** . Questo modo di ragionare sarà nel prosieguo sintetizzato mediante la sola scrittura dei numeri relativi alle corrispondenze, nella sequenza dipendente dal tipo di combinazione. Dopo questa premessa computiamo la funzione già esaminata:

Y = Sen x Cos x + Tang x

per due valori della variabile indipendente x = 3.5° e x = 37° sessagesimali.
In questo caso dobbiamo trasformare i valori sessagesimali in radianti, così come abbiamo mostrato all'inizio del paragrafo 1.9.2, impiegando la costante moltiplicativa k = .017453293.
Il programma si compone, nell'ordine, in base alle corrispondenze simboliche 10), 3), 11), 5), 12) e, per semplificarne la compilazione, seguendo le indicazioni del paragrafo 2.5.

```
CLS

INPUT " x " ; x

k = .017453293

Y = SIN ( x * k ) * COS ( x * k ) + TAN ( x * k )

PRINT " Y = " ; Y
```

```
                F5
x ? 3.5
Y = .1220973
                2F5
x ? 37
Y = 1.234185

                F5
```

2.7 Osservazioni in merito all'istruzione CLS

L'istruzione CLS, posta in testa al programma allo scopo di pulire lo schermo video, può essere omessa se si desiderano raccogliere in una sola schermata più elaborazioni numeriche di uno stesso programma. Naturalmente, in questo caso, è necessario prestare attenzione a non confondere risultati di elaborazioni precedenti con risultati dell'ultima elaborazione.

2.8 Esercizio di programmazione n° 5 (funzione trigonometrica elementare)

Computiamo la funzione Y = Sec x per quattro valori di x .2; .4; .6; .8 radianti, raccogliendo tutte e quattro le operazioni di INPUT ed i quattro valori di Y in una sola schermata.
Per evitare di trovare sullo schermo dati calcolati in precedenza dobbiamo avere l'accortezza di digitare sulla schermata CLS e premere 2F5, quindi:
Secondo quanto indicato al paragrafo 2.7 omettiamo l'istruzione CLS ed in base alla corrispondenza simbolica 13) scriviamo:

```
INPUT " x " ; x

Y = 1/ COS ( x )

PRINT " Y = " ; Y
```

```
                F5
x ? .2
Y =  1.020339
                2F5
x ? .4
Y =  1.085704
                2F5
x ? .6
Y =  1.211628
                2F5
x ? .8
Y =  1.435324
                F5
```

2.9 Modalità di titolazione e memorizzazione di un programma compilato

Dopo aver compilato un programma, ed averne controllato la correttezza mediante valutazione dei valori assunti dalla variabile indipendente, lo si può titolare e memorizzare in modo che risulti nella memoria del P.C. come documento di lavoro. La procedura di titolazione e memorizzazione è molto semplice e si avvale della lista di operazioni di gestione, già citata al paragrafo 1.1, che si evidenzia cliccando col mouse sulla scritta **File**.
A seguito del comando del mouse su File compare, tra le altre, la scritta **salva con nome...**, cliccando poi su questa scritta viene presentata sul video una particolare maschera in cui, dentro un apposito riquadro, è riportata la scritta: **Nome del file**, a questo punto si digita il nome che vogliamo assegnare al programma seguito da un punto e dalla dizione BAS come nell'esempio che segue:

Nome del file PROVA.BAS (invio)

dopo il comando invio compare in alto sullo schermo, al posto della scritta "senza titolo" la nuova dicitura " PROVA.BAS " .
Nell'eseguire questa operazione si deve tenere presente che il nome che vogliamo assegnare al programma non deve avere più di 8 caratteri; se digitiamo più di 8 caratteri e premiamo (invio) compare un riquadro nel quale è indicato " nome di file errato", per correggere dobbiamo premere il tasto ESC e modificare entro il numero di caratteri stabilito.
Una volta che il programma è titolato, sia che si proceda ad altro lavoro, sia che si ritenga di sospendere, si può memorizzarlo cliccando su **File** e di seguito su **Salva**.

2.10 Esercizio di programmazione n° 6 (funzione parametrica composta)

Costruiamo il programma per il calcolo della seguente funzione parametrica composta:

$$Y = \text{Cosec } x + A \text{ Cot } x$$

da computare per due valori del parametro A e per quattro valori della variabile indipendente .
Per A 1.5 e 3, per x .1; .2; .3; .4 .
E' utile, nel compilare il programma, commentare le diverse istruzioni per ricordare la sequenza degli eventi richiesti .
Digiteremo pertanto in base alle corrispondenze simboliche 14), 5), 3) e15):

```
CLS ' pulizia schermo

INPUT " A " ; A ' richiesta ingresso valore parametro

INPUT " x " ; x ' richiesta ingresso variabile indipendente

Y = 1/ SIN ( x ) + A * ( 1/ TAN ( x ) ) ' calcolo funzione parametrica

PRINT " Y = " ; Y ' comando visualizzazione dati variabile dipendente
```

```
                F5

A ? 1.5
x ? .1
Y = 24.96665

                2F5
A ? 1.5
x ? .2
Y = 12.43322
                2F5
A ? 1.5
x ? .3
Y = 8.232955
                2F5
A ? 1.5
x ? .4
Y = 6.115766
                2F5
A ? 3
x ? .1
Y =  39.91662
                2F5
A ? 3
x ? .2
Y = 19.83295
                2F5
A ? 3
x ? .3
Y = 13.08205
                2F5
A ? 3
x ? .4
Y = 9.6636

                F5
```

2.11 Sistema automatico per il calcolo di funzioni a campo fisso

L'esercizio del paragrafo precedente ha richiesto una procedura lunga e ripetitiva per ottenere i risultati voluti; infatti il valore del parametro A è stato digitato 8 volte e altre 8 volte sono stati digitati i valori di x. Si può comprendere quanto lavoro richiederebbe la computazione con più parametri estesa a un maggior numero dei valori della variabile indipendente.
Per semplificare la computazione delle funzioni viene in aiuto una semplice routine di programma che prevede, una volta fissato a priori il campo di escursione e l'entità degli incrementi della variabile indipendente, il computo automatico della Y = f (x) in tutto il campo stabilito.
Mostriamo la routine, completa di tutte le istruzioni necessarie per l'applicazione immediata, digitata e commentata per una generica funzione espressa simbolicamente.

```
INPUT " K" ; K  ' richiesta di ingresso del parametro

FOR  x = m  TO  n  STEP s  ' l'istruzione impone che  la  variabile indipendente (x) inizi il
                           ' calcolo assumendo il valore (m) , quando l'esecuzione del  programma giunge
                           ' all 'istruzione  NEXT, il programma ritorna all'istruzione FOR che incrementa x
                           ' del valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                           ' quando il valore di x ha raggiunto il valore (n)

Y =  f ( x ; K )  ' funzione parametrica simbolica da computare

PRINT "Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x  ' comanda il programma al ritorno automatico all 'istruzione FOR
```

L'applicazione del programma esposto trova riscontro pratico nel paragrafo seguente.

2.12 Esercizio di programmazione n° 7 (funzione parametrica esponenziale)

Visto il sistema automatico per il computo delle funzioni applichiamolo alla funzione parametrica esponenziale:

$$Y = e^{(kx)}$$

che ha nelle 16) e 3) le corrispondenze simboliche in Qbasic.
L'esercizio prevede che il parametro k assuma due valori: .3; .5 e che la variabile indipendente assuma dieci valori; da x = 3 a x = 21 ad intervalli di 2 unità .

Compiliamo il programma commentandolo:

```
CLS  ' pulizia dello schermo

INPUT " k" ; k  ' richiesta di ingresso del parametro

FOR  x = 3  TO  21  STEP 2  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                            ' valore (3) , quando l'esecuzione del programma giunge alla istruzione
                            ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                            ' valore (2) . Il ciclo si ripete con incrementi di (2) per arrestarsi
                            ' quando il valore di x ha raggiunto il valore (21)

Y = EXP ( k * x )  ' funzione parametrica da computare                                    &
```

```
PRINT "Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x  ' comanda il programma al ritorno automatico alla istruzione FOR fino al raggiungimento
        ' dell'ultimo valore previsto per la variabile indipendente.
```

```
                F5
k ? .3
Y = 2.459603
Y = 4.481689
Y = 8.16617
Y = 14.87973
Y = 27.11264
Y = 49.40246
Y = 90.01715
Y = 164.0219
Y= 298.8675
Y = 544.572
                2F5
k ? .5
Y = 4.481689
Y =12.18249
Y =33.11545
Y =90.01713
Y =244.6919
Y =665.1417
Y =1808.042
Y =4914.769
Y=13359.73
Y =36315.5
                F5
```

2.13 Sistema automatico per il calcolo di funzioni a campo variabile

La routine di calcolo illustrata nel paragrafo 2.11 può essere estesa per la computazione di funzioni nell'ambito di un campo variabile, in escursione ed entità degli incrementi, della x.
Mostriamo la routine, completa di tutte le istruzioni necessarie per l'applicazione immediata, digitata e commentata per una generica funzione espressa simbolicamente.

```
INPUT " K" ; K  ' richiesta di ingresso del parametro

INPUT " m " ; m  ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n  ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s  ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR x = m TO n STEP s  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                       '  valore (m) , quando l'esecuzione del programma giunge alla istruzione
                       ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                       ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                       ' quando il valore di x ha raggiunto il valore (n)

Y = f ( x ; K )  ' funzione parametrica da computare

PRINT "Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x  ' comanda il programma al ritorno automatico alla istruzione FOR
```

L'applicazione del programma esposto trova riscontro pratico nel paragrafo seguente.

2.14 Esercizio di programmazione n° 8 (funzione logaritmica composta)

Per esercitarci sulla routine di calcolo automatico a campo variabile, descritta nel paragrafo precedente, computiamo la seguente funzione logaritmica composta:

$$Y = \ln x \cdot \log x$$

Prima di procedere alla stesura del programma e necessario fare una importante osservazione relativa al campo di variabilità della x; la funzione che ci accingiamo a computare è del tipo logaritmico e pertanto non può essere calcolata né per x = 0, valore per il quale è infinitamente grande, né per valori negativi di x che sono estranei al campo di variabilità di questo tipo di funzione. Per evitare quindi che il programma porti ad una condizione di errore, denunciata dalla indicazione **"CHIAMATA DI FUNZIONE NON VALIDA"**, il valore di (x) deve essere, in questo esempio di calcolo, sempre maggiore di zero; **x > 0** .
Ciò premesso stabiliamo per il nostro esercizio il campo di variabilità nei seguenti limiti:
m = .1; n = 1 con incremento di x pari ad s = .2
In base alle corrispondenze simboliche 17); 3); 18) ed al programma del paragrafo 2.13, nel quale omettiamo la prima istruzione non essendo questo esercizio su di una funzione parametrica, compiliamo il nuovo programma:

```
CLS ' pulizia dello schermo

INPUT " m " ; m ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR x = m TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                      ' valore (m) , quando l'esecuzione del programma giunge alla istruzione
                      ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                      ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                      ' quando il valore di x ha raggiunto il valore (n)

Y = LOG ( x ) * LOG ( x ) / LOG ( 10 ) ' funzione logaritmica composta da computare

PRINT "Y = " ; Y ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione FOR
```

F5

```
m ? .1
n ? 1
s ? .2
Y = 2.302585
Y = .6295317
Y = .2086581
Y = 5.524965E-02
Y = 4.821035E-03
```

F5

I risultati che abbiamo ottenuto evidenziano, per la prima volta in un esercizio, le **notazioni scientifiche E-02**; **E-03**, esse indicano che le cifre poste alla loro sinistra devono essere divise rispettivamente per 100 e per 1000; la prima cifra vale 0.05524965 e la seconda vale 0.004821035.

In generale:

la notazione E - 0 n implica la divisione del numero alla sua sinistra per 10^n

la notazione E + 0 n implica la moltiplicazione del numero alla sua sinistra per 10^n

La semplicità e la rapidità di questo metodo di calcolo si commentano da sole, il computo della funzione può essere ripetuto a piacimento per qualsiasi campo di variabilità della x con l'incremento che può essere scelto in base alle necessità di definizione del calcolo; infatti più piccolo sarà il valore assegnato ad (s) maggiore sarà il numero dei punti calcolati, nell'ambito del campo imposto. Visti i risultati che abbiamo ottenuto con il sistema automatico di calcolo, gran parte degli esercizi a seguire saranno svolti secondo questa metodologia.

2.15 Come richiamare in video un programma precedentemente memorizzato

Se si vuole richiamare in video un programma, già memorizzato secondo quanto indicato nel paragrafo 2.9, si procede come segue:
Si clicca con il mouse su **File** e di seguito su **Apri.....** ; si ha la comparsa di un riquadro con la scritta **Nome del file:**
Si digita il nome del programma, ad esempio PROVA.BAS, e si preme (invio); si ha immediatamente sul video la comparsa della schermata del programma richiesto.

2.16 Come documentare un programma mediante stampante

Se si desidera documentare un programma mediante stampante, in modo da poterlo consultare in qualsiasi momento senza la necessità di operare con il P.C., si utilizza la sua presentazione sul video: o con la procedura indicata nel paragrafo precedente, o dopo averne ultimato la compilazione, lo si può quindi stampare semplicemente come sotto indicato:
Si clicca con il mouse su **File** e di seguito su **Stampa ....**; si ha la comparsa di un riquadro sul video con le scritte:

( ) Solo testo selezionato
( ) Finestra corrente
( ) Programma intero

si clicca con il mouse all'interno dell' ultima parentesi e si preme (invio); la stampante si mette in azione e tutto è fatto. La stampa che si ottiene è detta **listato del programma**.
Per una più completa documentazione di un programma è opportuno allegare al suo listato la stampa di un computo numerico relativo ad una sua applicazione, da ottenere con le modalità indicate nel paragrafo 1.4.2.2 o 1.4.2.3.

2.17 Esercizio di programmazione n° 9 (funzione di funzione)

Per esercitarci con le funzioni di funzioni proponiamoci il computo della funzione ciclometrica seguente:

$$Y = \text{Arcotang}\,(t)$$

dove (t) è a sua volta dipendente da un'altra funzione del tipo:

$$t = e^{x}$$

dove x è la sola variabile indipendente . Le due funzioni possono essere rappresentate con una sola espressione sostituendo alla lettera (t) della prima espressione l'esponenziale della seconda:

$$Y = \text{Arcotang} (e^{x})$$

Assumendo come valori del campo di variabilità della x m =.1 n = 8.1 con s = 2, mediante le corrispondenze simboliche 19) e 16) possiamo compilare il programma:

```
CLS ' pulizia dello schermo

INPUT " m " ; m  ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n  ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s  ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR x = m TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                      ' valore (m) , quando l'esecuzione del programma giunge alla istruzione
                      ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                      ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                      ' quando il valore di x ha raggiunto il valore (n)

Y = ATN ( EXP ( x ) ) ' funzione di funzione da computare

PRINT "Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x ' comanda il programma al ritorno automatico alla istruzione FOR
```

```
          F5
m ? .1
n ? 8.1
s ? 2
Y = .835315
Y= 1.448947
Y=1.554225
Y=1.568553
Y=1.570493
          F5
```

2.18 Esercizio di programmazione n° 10 (funzione di funzione)

Dato il diffuso impiego delle funzioni di funzioni proponiamo un secondo esercizio allo scopo di abituare il lettore alla manipolazione spedita di questo importante algoritmo.
Consideriamo la funzione ciclometrica:

$$Y = \text{Arcsen } t$$

nella quale la variabile (t) dipende a sua volta dalla funzione

$$t = b^{3}$$

nella quale la variabile (b) dipende infine da x secondo la funzione

$$b = \text{Cos } x$$

Le tre formule possono essere rappresentate da un solo algoritmo mediante le sostituzioni d'uso:

$$Y = \text{Arcsen } (\text{Cos } x)^3$$

La formula ottenuta ha il pregio di sintetizzare un considerevole numero di operazioni ma presenta difficoltà di implementazione in Qbasic se non si ha un poco di esperienza.
E' pertanto consigliabile, almeno in prima battuta, procedere alla compilazione del programma introducendo le tre funzioni separatamente mediante un metodo che sarà illustrato nei commenti delle diverse istruzioni.
Il metodo che impiegheremo, se in questo esercizio viene in aiuto al principiante, è in effetti uno strumento indispensabile quando ci si trovi a dover implementare, in routine di programma, funzioni di funzioni che sono proposte mediante un unico algoritmo molto complicato, tale da non essere trasformabile direttamente mediante le corrispondenze simboliche; è in questi casi che si rende necessario suddividere la funzione principale complicata in più funzioni semplici in modo da consentirne una agevole implementazione similmente a quanto ci accingiamo ora a fare.
Con l'aiuto delle corrispondenze simboliche 11); 2); 20) per m = .5; n = 1; s = .1 si compila il seguente programma:

```
CLS ' pulizia dello schermo

INPUT " m " ; m  ' richiesta di ingresso  dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n   ' richiesta di ingresso  dell'estremo superiore  del campo di variabilità di x

INPUT " s " ; s   ' richiesta di ingresso  dell'entità dell'incremento di x nell'ambito del campo

FOR  x = m  TO n  STEP s  ' impone che  la  variabile indipendente (x) inizi il calcolo assumendo il
                          ' valore (m) , quando l'esecuzione del programma giunge alla istruzione
                          ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                          ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                          ' quando il valore di x ha raggiunto il valore (n)

b = COS ( x )  ' si computa inizialmente la funzione  legata direttamente alla variabile indipendente x
               ' che nominiamo "prima"

t = b ^ 3      ' si computa di seguito la funzione legata direttamente alla "prima"
               ' che denominiamo "seconda"

Y = ATN ( t / SQR ( - t  *  t + 1 ) )  ' si computa infine la "terza" funzione legata direttamente alla
                                       ' seconda
PRINT "Y = " ;  Y  ' comando visualizzazione dati "terza" variabile dipendente

NEXT x  ' comanda il programma al ritorno automatico alla istruzione FOR
```

F5

```
m ? = .5
n ? =  1
s ? = .1
```

```
Y=.7421461
Y=.5970451
Y=.4638785
Y=.3449839
Y=.2425609

              F5
```

Non è superfluo ricordare che il programma è compilato in modo che il computo della funzione di funzione possa essere ripetuto per qualsivoglia campo di variabilità della variabile indipendente x, essendo flessibile la determinazione dei valori di (m) ed (n), e che nell'ambito del campo di variabilità di x si possono calcolare quanti valori della variabile indipendente Y siano necessari per ottenere una accurata analisi della funzione dipendente; ciò è infatti possibile mediante la definizione dell'incremento (s) .

2.19 Sistema automatico per il calcolo delle funzioni di più variabili in campi diversi

La routine di calcolo mostrata nel paragrafo 2.18 può essere estesa per la computazione di funzioni di più variabili nell'ambito di campi diversi, in escursione ed entità degli incrementi, delle variabili indipendenti.
Per semplicità di esposizione esamineremo il caso di funzioni a 2 variabili, l'estensione ad un maggior numero di queste si potrà poi ricavare da quanto verrà spiegato.
Ricordiamo la notazione simbolica esplicita di una funzione di due variabili:

$$Y = f (x1 ; x2)$$

per questa funzione il calcolo si sviluppa assegnando a x1 il primo valore voluto ed andando poi a computare i valori assunti dalla Y per tutti gli altri valori di x2, si ripete l'operazione per il secondo valore assegnato a x1 per tutti i valori che deve assumere x2, si procede in questo modo fino all'ultimo valore di x1 al quale associare sempre tutti i valori che deve assumere x2; pertanto il numero dei valori assunti dalla variabile dipendente è pari al prodotto del numero dei valori assunti da x1 ed i valori assunti da x2.
Se ad esempio per x1 si stabiliscono 10 valori e per x2 7 valori la variabile dipendente assumerà 70 valori.
Sulla base di quanto detto mostriamo la routine di calcolo commentata per una funzione a 2 variabili

```
INPUT " m1 " ; m1  ' richiesta di ingresso  dell'estremo inferiore del campo di variabilità di x1

INPUT " n1 " ;  n1  ' richiesta di ingresso  dell'estremo superiore  del campo di variabilità di x1

INPUT " s1 " ;  s1   ' richiesta di ingresso  dell'entità dell'incremento di x1 nell'ambito del campo

INPUT " m2 " ; m2  ' richiesta di ingresso  dell'estremo inferiore del campo di variabilità di x2

INPUT " n2 " ;  n2  ' richiesta di ingresso  dell'estremo superiore  del campo di variabilità di x2

INPUT " s2 " ;  s2   ' richiesta di ingresso  dell'entità dell'incremento di x2 nell'ambito del campo

FOR  x1 = m1  TO  n1  STEP s1  ' impone che  la  variabile indipendente (x1) inizi il calcolo
                               ' assumendo il valore (m1) , quando l'esecuzione del programma
                               ' giunge alla istruzione NEXT x1, il programma ritorna all'istruzione
                               ' FOR  che incrementa x1                                          &
```

```
FOR  x2 = m2  TO  n2  STEP s2  ' impone che  la  variabile indipendente (x2) inizi il calcolo
                               ' assumendo il valore (m2) , quando l'esecuzione del programma
                               ' giunge alla istruzione NEXT x2, il programma ritorna all'istruzione
                               ' FOR  che incrementa x2

Y =  f ( x1 ; x2 )  ' funzione  di due variabili da computare

PRINT "Y = " ;  Y  ' comando visualizzazione dati variabile dipendente

NEXT x2  ' comanda il programma al ritorno automatico alla istruzione FOR x2 = ...

NEXT x1  ' comanda il programma al ritorno automatico alla istruzione FOR x1 = ...
```

Come si vede la routine si sviluppa come abbiamo premesso; viene assegnato ad x1 il primo valore, e di seguito tutti i valori di x2 prima che x1 subisca il primo incremento, quando tutti i valori di x2 sono stati introdotti x1 si incrementa del primo valore pari ad s1 e nuovamente vengono assegnati di seguito tutti i valori di x2; il processo continua fino a quando x1 a seguito dei successivi incrementi raggiunge il valore di n1.

2.19.1 Esercizio di programmazione n° 11 (funzione di due variabili in automatico)

Questo esercizio si basa sulla routine automatica per il computo delle funzioni di due variabili spiegata nel paragrafo 2.19, viene computata la seguente funzione trigonometrica composta:

$$Y = \text{Sen } x1 + \text{Cos } x2$$

in due campi di variabilità così definiti:

per x1 da m1 = 0 a n1 = .2 radianti con s1 = .1
per x2 da m2 = 0 a n2 = .6 radianti con s2 = .2

Il primo campo comprende 3 valori della variabile indipendente x1, il secondo campo comprende 4 valori della variabile indipendente x2, per conseguenza la variabile dipendente assumerà 12 valori.
In base alle corrispondenze simboliche 10); 5); 11) si compila il seguente programma:

```
CLS  ' pulizia schermo

INPUT " m1 " ;  m1  ' richiesta di ingresso  dell'estremo inferiore del campo di variabilità di x1

INPUT " n1 " ;  n1   ' richiesta di ingresso  dell'estremo superiore  del campo di variabilità di x1

INPUT " s1 " ;  s1   ' richiesta di ingresso  dell'entità dell'incremento di x1 nell'ambito del campo

INPUT " m2 " ;  m2  ' richiesta di ingresso  dell'estremo inferiore del campo di variabilità di x2

INPUT " n2 " ;  n2   ' richiesta di ingresso  dell'estremo superiore  del campo di variabilità di x2

INPUT " s2 " ;  s2   ' richiesta di ingresso  dell'entità dell'incremento di x2 nell'ambito del campo

FOR  x1 = m1  TO  n1  STEP s1  ' impone che  la  variabile indipendente (x1) inizi il calcolo
                               ' assumendo il valore (m1) , quando l'esecuzione del programma
                               ' giunge alla istruzione NEXT x1, il programma ritorna all'istruzione
                               ' FOR  che incrementa x1
FOR  x2 = m2  TO  n2  STEP s2  ' impone che  la  variabile indipendente (x2) inizi il calcolo          &
```

```
                    ' assumendo il valore (m2) , quando l'esecuzione del programma
                    ' giunge alla istruzione NEXT x2, il programma ritorna all'istruzione
                    ' FOR  che incrementa x2

Y = SIN ( x1 ) + COS ( x2 ) ' funzione  di due variabili da computare

PRINT "Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x2  ' comanda il programma al ritorno automatico alla istruzione FOR x2 =

NEXT x1  ' comanda il programma al ritorno automatico alla istruzione FOR x1 =
```

```
                F5
m1 ? 0
n1  ? .2
s1  ? .1
m2 ? 0
n2  ? .6
s2  ? .2
Y=1
Y=.9800666
Y=.921061
Y=.8253356
Y=1.099833
Y=1.0799
Y=1.020894
Y=.925169
Y=1.198669
Y=1.178736
Y=1.11973
Y=1.024005
                F5
```

2.20 Esercizio di programmazione n° 12 (funzione iperbolica)

Gli esercizi di programmazione svolti finora hanno interessato tutte le funzioni elementari, 20) compresa; proseguiamo il lavoro computando la funzione iperbolica 23), tangente iperbolica, dato che questa esprime tanto il seno 21) quanto il coseno iperbolico 22) in quanto è data dal loro rapporto.
La tangente iperbolica è espressa in modo convenzionale da:

$$Y = \text{th } x$$

ci proponiamo di computarla nel campo di variabilità della x compreso tra .1 e .6 con incrementi di .1.
Compiliamo il programma con l'ausilio della corrispondenza simbolica 23) e con il sistema automatico a campo fisso illustrato nel paragrafo 2.11.

```
CLS  ' pulizia dello schermo

FOR  x = .1  TO  .6  STEP .1  ' impone che la variabile indipendente (x) inizi il calcolo assumendo
                              ' il valore ( .1 ) , quando l'esecuzione del programma giunge alla
                              ' istruzione  NEXT, il programma ritorna all'istruzione FOR che
                              ' incrementa x del valore (.1 ) . Il ciclo si ripete con incrementi di (.1)
                              ' per arrestarsi quando il valore di x ha raggiunto il valore ( .6 )

Y = ( EXP ( x ) - EXP ( -x ) ) / ( EXP ( x ) + EXP ( -x ) ) ' funzione iperbolica da computare          &
```

```
PRINT "Y = " ; Y   ' comando visualizzazione dati variabile dipendente

NEXT x  ' comanda il programma al ritorno automatico alla istruzione FOR fino al raggiungimento
        ' dell'ultimo valore previsto per la variabile indipendente.
```

```
                F5
Y =.099668
Y=.1973753
Y=.2913126
Y=.379949
Y=.4621172
Y=.5370496
                F5
```

2.21 Esercizio di programmazione n° 13 (funzione Sen x / x)

La prima delle funzioni speciali che andiamo a computare è la funzione fratta

$$Y = \frac{\text{Sen } x}{x}$$

questa funzione ha una particolarità; è indeterminata per x = 0 dato che per questo valore della variabile indipendente tanto il numeratore che il denominatore sono nulli. E' chiaro che se noi tentiamo il computo della funzione per x = 0 il Qbasic blocca l'esecuzione del calcolo indicando **"Chiamata di funzione illegale"** mentre sappiamo che il valore della funzione, per x che tende a 0, tende ad 1. E' pertanto indispensabile che il campo di variabilità della x non comprenda il valore 0; se si ha necessità di operare per tale valore lo si deve sostituire con un numero molto piccolo quale ad esempio .000001.
Ciò premesso possiamo compilare il programma in base alla corrispondenza simbolica 24) ed al sistema automatico a campo variabile per m = .000001 n = 2 s = .2 .

```
CLS 'pulizia dello schermo

INPUT " m " ; m  ' richiesta di ingresso dell'estremo inferiore del campo di variabilità di x

INPUT " n " ; n   ' richiesta di ingresso dell'estremo superiore del campo di variabilità di x

INPUT " s " ; s   ' richiesta di ingresso dell'entità dell'incremento di x nell'ambito del campo

FOR  x = m  TO n  STEP s  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                          ' valore (m) , quando l'esecuzione del programma giunge alla istruzione
                          ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                          ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                          ' quando il valore di x ha raggiunto il valore (n)
Y = SIN ( x ) / x  ' funzione da computare

PRINT "Y = " ; Y  ' comando visualizzazione dati variabile dipendente

NEXT x  ' comanda il programma al ritorno automatico alla istruzione FOR
```

```
                F5
m ? .000001
n ? 2
s ? .2
Y=1
```

```
Y=.9933466
Y=.9735457
Y=.9410706
Y=.8966949
Y=.8414707
Y=.7766989
Y=.7038923
Y=.624733
Y= .541026
```

F5

2.21.1 Osservazioni sulle funzioni fratte

Nell'esercizio precedente abbiamo preso delle precauzioni nel calcolo della funzione Sen x /x onde evitarne l'indeterminazione. In generale nel computare qualsiasi funzione fratta, dotata cioè di denominatore funzione della variabile indipendente, è indispensabile calcolare gli eventuali zeri del denominatore per escluderli opportunamente dal campo di variabilità, sia per evitare forme indeterminate, sia per evitare forme di infinito.

2.22 Esercizio di programmazione n° 14 (la funzione gaussiana)

La funzione gaussiana, che gioca un ruolo importante nel campo dell'analisi dei segnali, è una particolare funzione di funzione, esponenziale e parametrica, in cui l'esponente varia con il prodotto del parametro con il quadrato della variabile indipendente secondo l'espressione:

$$y = e^{-a x^2}$$

Nonostante la complessità della dipendenza di y dalla variabile indipendente la corrispondenza simbolica in Qbasic è molto semplice ed è espressa mediante la 25); corrispondenza secondo la quale ne eseguiamo il computo per un campo di variabilità della x compreso tra 1 e 5 con incrementi a passi unitari e parametro a = .01.

```
CLS ' pulizia dello schermo

a = .01 ' valore del parametro

FOR  x = 1 TO 5 STEP 1 ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                       ' valore (1) , quando l'esecuzione del programma giunge alla istruzione
                       ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                       ' valore (1) . Il ciclo si ripete con incrementi di (1) per arrestarsi
                       ' quando il valore di x ha raggiunto il valore (5)

Y = EXP ( - a  *  x ^ 2 )   ' funzione gaussiana da computare

PRINT "Y = " ;  Y   ' comando visualizzazione dati variabile dipendente

NEXT x  ' rimanda all'istruzione  FOR  per un successivo incremento di x
```

F5

```
Y=.9900498
Y=.9607894
Y=.9139312
Y=.8521438
Y=.7788008
```

F5

E' opportuno osservare che i metodi di computo automatico a campo fisso quando, come in questo caso, hanno il valore dell'incremento (s) = 1 possono essere compilati omettendo la notazione **STEP 1.**

2.23 Esercizio di programmazione n° 15 (il fattoriale)

L'esercizio relativo alla funzione fattoriale, così come lo andiamo ora a sviluppare, è molto semplice e di poca eleganza formale; un metodo automatico è applicabile ma di scarso interesse pratico. La funzione fattoriale è espressa da:

$$Y = x!$$

procediamo al computo di un unico valore per x = 6 mediante la corrispondenza simbolica 26):

```
CLS

Y = 1 * 2 * 3 * 4 * 5 * 6

PRINT "Y = " ; Y
```

F5

```
Y = 720
```

F5

2.24 Esercizio di programmazione n° 16 (sommatoria algebrica)

La sommatoria algebrica ricorre sovente nei problemi tecnici e matematici, la funzione che la caratterizza è la seguente

$$Y = \sum_{x=p}^{x=n} x$$

essa ha lo scopo di eseguire il computo della somma di un certo numero di valori assunti dalla variabile indipendente x; se ad esempio p = 5 e n = 9 e l'incremento di x è unitario, per computare Y dobbiamo sommare i valori 5 + 6 + 7 + 8 + 9 ottenendo Y = 35. L'operazione di per se è banale ma il concetto di sommatoria è generalmente applicato a funzioni complicate e pertanto il suo computo diventa impegnativo.
Se vogliamo ora implementare la funzione in un programma in Qbasic e computarla per p =1, n = 5 con incrementi a passi di s =.5, dobbiamo ricorrere alla corrispondenza simbolica 27) che commenteremo a fianco delle singole istruzioni:

```
CLS ' pulisce lo schermo

INPUT " p " ; p ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " n " ; n  ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s " ; s  ' richiesta di ingresso valore dell'incremento di x

FOR x = p  TO n  STEP s  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                         ' valore (p) , quando l'esecuzione del programma giunge alla istruzione
                         ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                         ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                         ' quando il valore di x ha raggiunto il valore (n)                    &
```

```
K = x  ' uguaglianza di appoggio per l'istruzione successiva ( non necessaria in questo caso)

Y = Y + K  ' somma al valore di  Y in memoria  ( uguale a zero all'inizio) il primo valore di K , e
           ' sostituisce in memoria il valore di Y=0  con il valore di Y=K , dopo la prima istruzione
           ' NEXT , x si incrementa di ( s ) e per conseguenza si incrementa anche K; il valore
           ' di Y in memoria si aggiorna sommandosi al nuovo valore di K, le somme si ripetono
           ' con ulteriori aggiornamenti del valore di Y in memoria fino a quando x raggiunge il
           ' valore dell'estremo superiore del campo di variabilità

NEXT  x  ' rimanda all'istruzione FOR per un successivo incremento di x

PRINT " Y = " ;  Y  ' comanda la presentazione del valore finale di    Y
```

```
                F5
p ? 1
n ? 5
s ? .5
Y =  27
                F5
```

2.25 Esercizio di programmazione n° 17 (sommatoria algebrica progressiva)

Il risultato dell'esercizio precedente ci ha permesso di computare la sommatoria complessiva dei valori assunti da x nel campo di variabilità p = 1 n = 5 con incrementi di x pari a s =.5 definiti dalla seguente serie numerica 1; 1.5; 2; 2.5; 3; 3.5; 4; 4.5; 5. A volte è necessario conoscere anche i valori che la variabile dipendente Y assume mano a mano che la sommatoria avanza secondo gli incrementi assegnati alla x. Per far ciò è necessario modificare il programma del paragrafo 2.24 come segue (le modifiche sono indicate con il simbolo §):

```
CLS ' pulisce lo schermo

INPUT " p " ;  p  ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " n " ;  n   ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s " ;  s   ' richiesta di ingresso valore dell'incremento di x

FOR  x = p   TO  n   STEP  s   ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                               ' valore (p) , quando l'esecuzione del programma giunge alla istruzione
                               ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                               ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                               ' quando il valore di x ha raggiunto il valore (n)

K = x  ' uguaglianza di appoggio per l'istruzione successiva ( non necessaria in questo caso)

Y = Y + K  ' somma al valore di  Y in memoria  ( uguale a zero all'inizio) il primo valore di K , e
           ' sostituisce in memoria il valore di Y=0  con il valore di Y=K , dopo la prima istruzione
           ' NEXT  x si incrementa di ( s ) e di conseguenza si incrementa anche K; il valore
           ' di Y in memoria si aggiorna sommandosi al nuovo valore di K, le somme si ripetono
           ' con ulteriori aggiornamenti del valore di Y in memoria fino a quando x raggiunge il
           ' valore dell'estremo superiore del campo di variabilità

PRINT " x = " ; x , " Y = " ;  Y   ' § con questa istruzione si impone la presentazione dei valori di x
                                   ' dal primo all'ultimo affiancati ai valori progressivi assunti da Y,
                                   ' l'ultimo valore di  Y  rappresenta la sommatoria totale

NEXT  x  ' rimanda all'istruzione FOR per un successivo incremento di x                    &
```

```
' § è stata omessa l'istruzione PRINT " Y = " ; Y
```

F5

```
p ? 1
n ? 5
s ? .5
x = 1       Y = 1
x = 1.5     Y = 2.5
x = 2       Y = 4.5
x = 2.5     Y = 7
x = 3       Y = 10
x = 3.5     Y = 13.5
x = 4       Y = 17.5
x = 4.5     Y = 22
x = 5       Y = 27
```

F5

2.26 Esercizio di programmazione n° 18 (sommatoria di funzione)

La sommatoria di funzione ha la stessa struttura della sommatoria algebrica, differisce da questa per la presenza di una funzione al posto della variabile indipendente; con la sommatoria di funzione si computa la somma di un certo numero di valori assunti dalla variabile dipendente quando la variabile indipendente varia nel campo assegnato ad incrementi stabiliti.
L'espressione che segue sintetizza il concetto:

$$Y = \sum_{x=p}^{x=n} f(x)$$

L'espressione mostra la sommatoria dei valori assunti da una generica funzione f (x) quando x assume i valori del proprio campo di variabilità ad iniziare da x = p per terminare con il valore x = n; nell'algoritmo non è specificata l'entità dell'incremento della variabile indipendente che viene invece definita nell'ambito della compilazione della routine di programma.
Iniziamo l'esercizio di programmazione scegliendo come funzione da utilizzare per la sommatoria l'esponenziale

$$K = e^{x}$$

nella quale la variabile dipendente è indicata con K per lasciare ad Y la rappresentanza della funzione sommatoria; vogliamo computare Y nel campo della variabile indipendente definito tra x = .1 ed x = 10 con incrementi dell'ordine di .2; con l'ausilio delle corrispondenze simboliche 28) e 16) compiliamo e commentiamo il nostro programma:

```
CLS ' pulisce lo schermo

INPUT " p " ; p ' richiesta di ingresso valore inferiore del campo di variabilità della x

INPUT " n " ; n ' richiesta di ingresso valore superiore del campo di variabilità della x

INPUT " s " ; s ' richiesta di ingresso valore dell'incremento di x

FOR x = p TO n STEP s ' impone che la variabile indipendente (x) inizi il calcolo assumendo il &
```

```
                    ' valore (p) , quando l'esecuzione del programma giunge alla istruzione
                    ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                    ' valore (s) . Il ciclo si ripete con incrementi di (s) per arrestarsi
                    ' quando il valore di x ha raggiunto il valore (n)

K = EXP ( x ) ' funzione esponenziale soggetta alla sommatoria

Y = Y + K ' somma al valore di Y in memoria ( uguale a zero all'inizio) il primo valore di K , e
          ' sostituisce in memoria il valore di Y=0 con il valore di Y=K , dopo la prima istruzione
          ' NEXT , x si incrementa di ( s ) e di conseguenza si incrementa anche K; il valore
          ' di Y in memoria si aggiorna sommandosi al nuovo valore di K, le somme si ripetono
          ' con ulteriori aggiornamenti del valore di Y in memoria fino a quando x raggiunge il
          ' valore dell'estremo superiore del campo di variabilità

NEXT x ' rimanda all'istruzione FOR per un successivo incremento di x

PRINT " Y = " ; Y  ' comanda la presentazione del valore finale di  Y
```

```
                F5
p ? .1
n ? 10
s ? .2
Y =  109943.5
                F5
```

L'esercizio ha portato, come era previsto, al computo della sommatoria complessiva di funzione; se si desiderano computare i valori progressivi della sommatoria si devono apportare al programma le semplici modifiche indicate in precedenza al paragrafo 2.25.

2.27 Come copiare un programma o parte di esso

Si rende necessario talvolta copiare un programma o parte di esso per ottenere un altro programma diverso dal primo per alcune varianti; ne è un esempio il caso citato alla fine del paragrafo precedente che propone la modifica del programma esistente. Se però si ha la necessità di tenere in memoria il programma originale è indispensabile farne una copia da modificare.
L'operazione di copiatura è semplice e consiste nelle seguenti operazioni:

- Si preme il tasto delle maiuscole e contemporaneamente con i tasti di direzione si evidenziano il programma o le righe di questo da copiare
- Si preme il tasto **CTRL** contemporaneamente al tasto **INS**
- Si passa ad una schermata " senza nome"
- Si preme il tasto delle maiuscole contemporaneamente al tasto **INS**

Dopo quest'ultima azione si ha sullo schermo la comparsa del programma o della parte di questo che è stata copiata; naturalmente il testo originale è rimasto integro.

2.28 Esercizio di programmazione n° 19 (la parabola)

Per questa esercitazione prendiamo la parabola come funzione rappresentativa delle operazioni matematiche che caratterizzano le funzioni della geometria analitica (iperbole esclusa); nella funzione parabola infatti sono presenti prodotti, elevamenti a potenza, somme e parametri .
La funzione parabola è rappresentata dall'espressione:

$$Y = a x^2 + bx + c$$

in cui le lettere a; b; c sono i parametri della funzione.

Procedendo al computo di questa funzione mediante la corrispondenza simbolica 30), assumendo il campo di variabilità della x compreso tra -5 e +5 con incrementi unitari, ed i seguenti parametri a = 1; b = 1; c = -2, con l'ausilio dell'istruzione di presentazione contemporanea di x e Y, si ha:

```
CLS ' pulizia dello schermo

FOR x= -5 TO 5  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                ' valore (-5) , quando l'esecuzione del programma giunge alla istruzione
                ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                ' valore (1) . Il ciclo si ripete con incrementi di (1) per arrestarsi
                ' quando il valore di x ha raggiunto il valore (5)

Y= x^2 + x - 2  ' funzione parabola da computare

PRINT " x= " ; x , "Y= " ; Y  ' comando visualizzazione dati variabile indipendente associati ai
                              ' corrispondenti valori variabile dipendente

NEXT x ' rimanda all'istruzione FOR per un incremento successivo della x
```

F5

```
x = -5    Y =  18
x = -4    Y =  10
x = -3    Y =  4
x = -2    Y =  0
x = -1    Y = - 2
x =  0    Y = - 2
x =  1    Y =  0
x =  2    Y =  4
x =  3    Y = 10
x =  4    Y = 18
x =  5    Y = 28
```

F5

I risultati di questo esercizio mostrano una caratteristica particolare della funzione parabola, caratteristica per cui, in dipendenza dei valori dei parametri, la variabile dipendente si azzera due volte nel campo di variabilità della x; nel nostro caso gli zeri di Y si ottengono per i valori x = - 2 ed x = 1.
Come è noto questi due valori di x sono le soluzioni dell'equazione di 2° grado che si ottiene dalla funzione parabola uguagliandola a zero.

2.29 Esercizio di programmazione n° 20 (l' iperbole)

La computazione della funzione iperbole richiede attenzione particolare dato che essa è infinitamente grande per x = 0. E' chiaro che se noi tentiamo il computo della funzione per x = 0 il Qbasic blocca l'esecuzione del calcolo indicando " **Divisione per zero**". E' pertanto indispensabile che il campo di variabilità della x non comprenda valori tanto piccoli tali da portare il computer a denunciare tale condizione.
Ciò premesso possiamo compilare il programma in base alla corrispondenza simbolica 32) ed al sistema automatico a campo fisso per m = .1 n = 1.91 s = .2 .

```
CLS ' pulizia dello schermo

FOR x= .1 TO 1.91 STEP .2 ' impone che la variabile indipendente (x) inizi il calcolo assumendo  &
```

```
                    ' il  valore (.1) , quando l'esecuzione del programma giunge alla istruzione
                    ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                    ' valore (.2) . Il ciclo si ripete con incrementi di (.2) per arrestarsi
                    ' quando il valore di x ha raggiunto il valore (1.9)

Y = 1 / x   ' funzione iperbole da computare

PRINT " Y = " ; Y   ' comando visualizzazione dati variabile dipendente

NEXT x   ' rimanda all'istruzione FOR per un incremento successivo della x
```

F5

```
Y=10
Y=3.333333
Y=2
Y=1.428571
Y=1.111111
Y=.9090909
Y=.7692307
Y=.6666666
Y=.5882353
Y=.5263157
```

F5

2.30 Esercizio di programmazione n° 21 (operatori aritmetici relazionali)

Gli operatori aritmetici relazionali 8) e 9) introducono importanti ampliamenti delle capacità operative del sistema di calcolo in generale; mostriamo di seguito un piccolo esempio di ciò che si può fare con essi, lasciando al lavoro seguente ed alle capacità inventive del lettore che avrà maturato esperienza, le innumerevoli applicazioni possibili.
Riprendiamo allo scopo l'esercizio n° 1 relativo al calcolo della funzione:

$$Y = (x)^{1/2}$$

sviluppato per tre valori della variabile indipendente x = 3.75; x = 7; x = 12, un campo di variazione compreso tra i valori positivi 3.75 e 12; se il campo di variabilità della x fosse stato compreso tra - 3.75 e + 12 l'esercizio non sarebbe stato possibile, nell'ambito dei numeri reali, per l'estremo inferiore del campo, essendo esso un numero negativo. Ora con l'ausilio degli operatori aritmetici relazionali è possibile risolvere **"formalmente"** la funzione per qualsiasi campo di variabilità della x; ottenendo valori reali di Y per valori positivi di x e valori immaginari di Y per valori negativi di x.
Per giungere al risultato voluto è anzitutto necessario trasformare la funzione in oggetto mediante l'imposizione del valore assoluto 7) sotto radice:

$$Y = (| x |)^{1/2}$$

che consente il calcolo della funzione indipendentemente dal segno di x; ora mediante le corrispondenze simboliche 1) e 7) si può scrivere la funzione radice in valore assoluto della x come segue:

Y = SQR (ABS (x))

a questo punto non resta che compilare il nuovo programma che utilizza gli operatori relazionali e le istruzioni ad essi associate, istruzioni che saranno opportunamente commentate nel programma

stesso, per il computo della funzione nel campo di variabilità della x compreso tra -5 e + 5; i risultati attesi sono numeri reali e numeri immaginari, quest'ultimi distinti dai primi dal simbolo **j**:

```
CLS ' pulizia dello schermo

FOR  x = -5  TO  +5  ' impone che la variabile indipendente (x) inizi il calcolo assumendo il
                     ' valore ( -5 ) ,quando l'esecuzione del programma giunge alla istruzione
                     ' NEXT, il programma ritorna all'istruzione FOR che incrementa x del
                     ' valore ( 1 ) . Il ciclo si ripete con incrementi di ( 1 ) per arrestarsi
                     ' quando il valore di x ha raggiunto il valore ( + 5 )

Y = SQR ( ABS ( x ) )  ' funzione  radice con x in valore assoluto


IF  x < 0  THEN PRINT " j " ; Y  ' l'istruzione impone che se x è inferiore a zero venga stampato ,
                                 ' prima del valore numerico della variabile dipendente ,il simbolo
                                 ' j per indicare che è stata sviluppata (simbolicamente) una radice
                                 ' quadrata di un numero negativo e che il risultato è un numero
                                 ' immaginario puro

IF x >= 0  THEN PRINT ; Y    ' l'istruzione impone che se x è maggiore od uguale a zero il valore
                             ' numerico della variabile dipendente venga stampato in modo
                             ' normale dato che esso è il risultato della radice quadrata di un
                             ' numero positivo

NEXT x  ' comanda il ritorno all'istruzione FOR per gli incrementi successivi.
```

```
                F5
j 2.236068
j 2
j 1.732051
j 1.414214
j 1
0
1
1.414214
1.732051
2
2.236068

                F5
```

I risultati mostrano che la funzione in esame ha fornito cinque valori immaginari e sei valori reali di Y come la teoria prevede.
L'esercizio che abbiamo condotto si basa, sia sugli operatori aritmetici relazionali, sia sulle istruzioni **IF** x > a **THEN** ..... che sono estesamente trattate nei manuali del Qbasic; non rientrando negli obiettivi di questo testo un approfondimento su tale argomento si rimanda il lettore ai riferimenti bibliografici riportati nelle ultime pagine.

2.31 Sulle funzioni a due valori

Negli esercizi che abbiamo svolto non sono state elaborate numericamente le funzioni a due valori 31); 33); 34) relative alla geometria analitica; dato che per esse è significativa soltanto la presentazione grafica, ci riserviamo di trattarle in tal modo nel successivo capitolo 3.