

Eingabe / Verarbeitung / Ausgabe / Kommentar

REM 'Kommentar	Alles was nachfolgt wird nicht beachtet.
INPUT "Eingabe: ", Variable\$	Daten von Tastatur einlesen bis Return gedrückt wird.
PRINT "Text"; Variable;	Ausgabe von Texten und Variablen.
PRINT	Endet die Zeile mit einem Strichpunkt, dann keine neue Zeile.
(LET) X = X + 7 * Y	Bei Berechnungen gilt Punkt vor Strich. (LET darf weggelassen werden)
CLS	Löscht den Ganzen Bildschirm. (Farbe kann mit COLOR variiert werden)
COLOR Vordergrund, Hintergrund	Legt Fest mit welchen Farben ab sofort geschrieben wird. (nur bei SCREEN 0)
0: schwarz 1: blau 2: grün 3: cyan 4: rot 5: lila 6: braun 7: hell grau 8: grau 9: hell blau 10: hell grün 11: hell cyan 12: hell rot 13: rosa 14: gelb 15: weiß	
LOCATE Y, X (CSRLIN)	Cursor auf bestimmte Position setzen (screen 0) Y: 1 - 23 X: 1 - 80
PRINT USING "+###,##";Variable	Formatierte Ausgabe: + Vorzeichen / , Tausender Stellen / . Kommastelle
INPUT\$(Anzahl)	Lieft eine bestimmte Anzahl von Zeichen von der Tastatur ein.
INKEY\$	Gibt zurück welche Taste gedrückt worden ist.
Einige ASCII -Codes:	
218 194 191 201 203 187 219 176 220 177 223 178 221 222 196 205 195 197 180 204 206 185 179 186 192 193 217 200 202 188	Enter: 13 "M" "S" Entf ESC: 27 "K" 8 <-- Tab: 9 "H" "R" Einfg Space: 32 "P" "G" Pos1 Beispiel: CHR\$(27) "O" Ende CHR\$(0) + "M"

Variablen / Felder

INTEGER %	Ganzzahl (16 Bit) -32768 bis 32767
LONG &	Ganzzahl (32 Bit) -2.147.483.648 bis 2.147.483.647
SINGLE (standart) !	Fließkommazahl (32 Bit) bis zu 7 Nachkommastellen
DOUBLE #	Fließkommazahl (64 Bit) bis zu 16 Nachkommastellen (1,7977 D+308)
STRING \$	Text (variabler Länge)
STRING * n%	Text der Länge n%
DIM Variable AS INTEGER	Variable einem Bestimmten Typ zuordnen.
DIM Feld(40) ' = 0-40	Erzeugen von Arrays .
DIM Feld(1 TO 10, 5 TO 8)	Benutzen der Felder: Feld(7, 5) = 10
REDIM Feld(20) (AS INTEGER)	Zum ändern der Größe eines Arrays. (Achtung: Einträge werden gelöscht)
DEFINT DEFLNG DEFSNG	Alle Variablen ohne Endung, die nicht mit DIM Deklariert werden, sind standartmäßig vom Typ SINGLE . Dieser Befehl ändert den Standarttyp
DEFDBL DEFSTR	
OPTION BASE Zahl	Legt die unter Standartgrenze für Datenfeldindizes fest. (0 oder 1)
TYPE Spieler Name AS STRING * 30 Punkte AS INTEGER	Um Variablen aus verschiedenen Datentypen zusammen zu fassen kann man mit dem TYPE -Befehl sie zu einem neuen Datentyp zusammenfassen. Variablen dieses Typs werden ganz normal mit DIM dekariert:
END TYPE	
CONST Konstantenname = Wert	Der Wert der Konstanten darf nicht mehr geändert werden.

Abfragen (IF, SELECT CASE)

IF Bedingung THEN Befehl	Der Befehl wird nur dann ausgeführt, wenn Bedingung erfüllt wird.
IF Bedingung THEN	Wenn hinter dem THEN kein Befehl steht, dann führt er alle Befehle aus, bis zum ersten ELSEIF, ELSE oder END IF.
ELSEIF Bedingung THEN	Die Bedingung beim ELSEIF wird nur dann kontrolliert, wenn alle vorherigen Bedingungen nicht erfüllt worden sind.
ELSE	Wenn keine der Abfragen erfolgreich war, dann wird der Teil hinter dem ELSE ausgeführt. (wenn vorhanden)
END IF	Solche IF-Blöcke müssen mit einem END IF abgeschlossen werden.
SELECT CASE Variable	Wenn der Zustand einer Variablen den Auszuführenden Programm-Code bestimmt, dann ist SELECT CASE übersichtlicher als eine IF-Abfrage. Bei Vergleichen muß der Befehl IS vorangestellt werden. Sollte keine der Zustände auf die Variable zutreffen, dann wird der Teil hinter dem CASE ELSE ausgeführt. (wenn vorhanden)
CASE 1, 2, 4 TO 7	
CASE IS >= 8	
CASE ELSE	
END SELECT	

Bedingungen Zum Vergleichen von Variablen. Wird in Schleifen und Abfragen verwendet

Variable1 = Variable2	gleich	Mehrere Bedingungen mit einander Verknüpfen:							
Variable1 <> Variable2	ungleich	1.Bed	2.Bed	AND	OR	XOR	EQV	IMP	F: Bed nicht erfüllt
Variable1 < Variable2	kleine als	F	F	F	F	F	T	F	
Variable1 <= Variable2	kleiner gleich	F	T	F	T	T	F	F	T: Bed erfüllt
Variable1 > Variable2	größer als	T	F	F	T	T	F	T	
Variable1 >= Variable2	größer gleich	T	T	T	T	F	T	F	

Schleifen (FOR, DO ... LOOP)

FOR i = 1 TO 10 NEXT i	Wird verwendet, wenn die Anzahl der Schleifendurchläufe bekannt ist. Die Variable (hier i) wird bei jedem Schleifendurchlauf um eins erhöht.
FOR i = 10 TO 4 STEP -2	Mit STEP kann man die Veränderung der Variablen steuern. (hier wird zu i bei jedem Schleifendurchlauf -2 dazu addiert)
DO LOOP UNTIL Bedingung	Der Programmteil zwischen DO und LOOP wird solange ausgeführt, bis die Bedingung erfüllt wird. Die Schleife wird mindestens einmal durchlaufen, bevor das erste mal die Bedingung überprüft wird.
DO UNTIL Bedingung LOOP	Hier wird die Bedingung gleich zu Beginn überprüft, es kann also passieren, daß die Schleife nie ausgeführt wird.
DO LOOP WHILE Bedingung	DO WHILE Bedingung LOOP Solange die Bedingung erfüllt wird, wird die Schleife fortgesetzt.
EXIT FOR EXIT DO	Die jeweilige Schleife sofort verlassen.

Operatoren für Zahlen Es gilt Punkt vor Strich. Es können Klammern gesetzt werden.

+ - * /	Addition / Subtraktion / Multiplikation / Division
SQR (Zahl) 2^8	Wurzelziehen / Exponentialrechnung (2 hoch 8)
SIN (Zahl) COS (Zahl) TAN (Zahl) ATN (Zahl)	Sinus / Cosinus / Tangens (Bogenmaß = Winkel * PI / 180) Arcustangens (Ergebnis in Bogenmaß) PI = 3.1415926536
126 MOD 10	Gibt den Rest einer Division zurück. (Beispiel: 126 / 10 = 12 Rest 6)
126 \ 10	Führt Division durch und schneidet Nachkommastellen ab.
ABS (Zahl)	Bildete den Absolutwert. (negative Zahlen werden Positiv)
SGN (Zahl)	Negativ -> -1 Null -> 0 Positiv -> +1
INT (Zahl) FIX (Zahl)	Rundet auf die nächste Ganzzahl ab. / Schneidet Nachkommastellen ab.
CINT (Zahl)	Rundet eine Zahl. (Beispiel: CINT(1.49) -> 1 CINT(1.5) -> 2)

Strings

Text\$ = Text1\$ + Text2\$	Zwei Texte zu einem neuen Text verbinden.
Text\$ = STR\$(Zahl)	Wandelt eine Zahl in einen String um.
Zahl = VAL (Text\$)	Wandelt einen String in eine Zahl um.
Zeichen\$ = CHR\$(Zahl)	Wandelt Zahl nach dem ASCII-Code in ein Zeichen um.
Zahl = ASC (Text\$)	Ermittelt den ASCII-Code eines Zeichens.
Text1\$ = MID\$(Text2\$, 4, 2)	Gibt einen Teil des Textes zurück. (hier ab der 4. Position 2 Zeichen)
Text1\$ = MID\$(Text2\$, 4)	Gibt einen Teil des Textes zurück. (hier alle Zeichen ab der 4. Position)
Text1\$ = LEFT\$(Text2\$, 3)	Gibt die ersten Zeichen des Textes zurück. (hier die ersten 3 Zeichen)
Text1\$ = RIGHT\$(Text2\$, 3)	Gibt die letzten Zeichen des Textes zurück. (hier die letzten 3 Zeichen)
Space\$(Anzahl)	Erzeugt einen String aus der gewünschten Anzahl von Leerzeichen.
String\$(Anzahl, Zeichen\$)	Erzeugt einen String aus der gewünschten Anzahl des Zeichens\$.
Zahl = LEN (Text\$)	Gibt die Länge eines Textes als Zahlenwert zurück.
LTRIM\$(Text\$) RTRIM\$(Text\$)	Schneidet Links bzw. Rechts vom Texten die Leerzeichen ab.
LCASE\$(Text\$) UCASE\$(Text\$)	Wandelt alle Zeichen eines Textes in Klein- bzw. Großbuchstaben um

Timer / Zufallszahlen

TIMES\$ DATE\$	Geben Datum bzw. Uhrzeit als String zurück. (mm-tt-jjjj hh:mm:ss)
sek = TIMER	Gibt die Sekunden seit Mitternacht zurück.
SLEEP sek	Pause für angegebene Sekundenzahl. (nur ganze Sekunden)
(RANDOMIZE Zahl) RANDOMIZE TIMER Zahl% = INT(RND * 10) + 5 Zahl = RND * 10 + 5	Errechnet eine zufällig wirkende Zahlenfolge aus der übergebenen Zahl. INT(RND * (max-min+1)) + min -> zufällige Zahl zwischen 5 und 5 + 10 - 1 (= 14) -> zufällige Zahl zwischen 5 und 14,9999 (RND -> Zahl zwischen 0 und 1)

SUBs und Funktionen

DECLARE SUB Name (.....)	Deklariert eine Sub. (Das Unterprogramm bekannt machen)
SUB Name (Feld(), Var2 AS ...) END SUB	Der Programmcode der Sub kommt zwischen diese Zeilen. Die SUB kann nur auf Variablen vom Hauptprogramm zugreifen, die ihm übergeben werden oder als SHARED deklariert werden.
DIM SHARED Name (AS ...)	Alle SUBs können auf so im Hauptprogramm deklarierte Variablen zugreifen.
STATIC name (AS ...)	Nach Beendigung der SUB wird eine so deklarierte Variable nicht gelöscht.
CALL Name (var1, var2,)	Aufruf einer SUB. Wenn in der Declaration Parameter verlangt werden, dann müssen diese beim Aufruf übergeben werden.

Funktionen werden genauso behandelt wie SUBs, nur dass sie einen Wert ans aufrufende Programm zurückgeben.

DECLARE FUNCTION Name%()	Die Endung am Funktionsnamen gibt die Art des Rückgabewertes an. hier: INTEGER (siehe Variablen)
Name% = Variable%	In der Funktion muß man dem Funktionsnamen den Rückgabewert zuweisen.
Variable = Name% (Var1, Var2)	Aufruf einer Funktion. Wenn Parameter verlangt werden, dann müssen sie in Klammern hinter dem Funktionsnamen übergeben werden.
EXIT SUB EXIT FUNCTION	Bewirkt das sofortige Verlassen eines Unterprogrammes.

Beenden

END	Beendet das Programm und schließt alle Dateien.
SYSTEM	Schließt alle Dateien und kehrt zum Betriebssystem zurück.

Sound

BEEP	Gibt einen Signalton über den Lautsprecher aus.
Die Befehle SOUND und PLAY geben ihre Töne über den PC-Lautsprecher wieder.	
SOUND Frequenz, 18.2	Ton in angegebener Frequenz (37 - 32767) und Länge. (18,2 = 1 Sekunde)
PLAY "....."	Oktave: O4 (0 - 6) eine Oktave > höher / < tiefer Tohn: CDEFGAB oder N37 (1 - 64) (-> o3c) Halbtohn: höher: C+ D# tiefer: E- Geschw.: L4 (1 - 64) -> L1 = ganze Note 1,5 mal länger spielen: F. Pause: P8 (1 - 64) L2 = halbe Note ... Tempo: T144 (32 - 255) Noten: Normal MN Fließend ML Gehackt MS

Grafik

SCREEN Modus (, ,A, V)	A-Seite: Bearbeitungsseite V-Seite: Anzeigeseite
0 -> Textmodus (X , Y) Seiten	9 -> verschoben 0 - 1
1 -> 4 Farben (318, 199)	10 -> verschoben schwarz/grau/weiß blinkend/weiß
2 -> schwarz/weiß (639, 199)	11 -> schwarz/weiß (639, 479)
7 -> 16 Farben (318, 199) 0 - 7	12 -> 16 Farben (639, 479 (463))
8 -> 16 Farben (639, 199) 0 - 3	13 -> 247 Farben (318, 199 (191))
..... STEP (X, Y)	Beispiel: LINE (20 ,20) - STEP(30, 40) = LINE (20, 20) - (50, 60)
PSET (X, Y), Farbe	Einen Punkt zeichnen.
LINE (X, Y) - (X, Y), (Farbe, BF)	Linie Zeichnen. B: Rechteck BF: Ausgefülltes Rechteck
CIRCLE (X,Y),Radius,(Farbe,A,B)	Kreis zeichnen. A: Anfangswinkel B: End-Winkel des Kreisbogens
PAINT (X,Y), Füllfarbe, Randfarbe	Fläche ausmalen. (begrenzt durch die Randfarbe)
DRAW "....."	H U E Bsp.: U10 -> 10 nach unten Ausfüllen: P füllfarbe, randfarbe L R Drehen in Grad: TA15 (gegen Uhrzeigersinn) nicht zeichnen: B... Bsp.: BU10 G D F Farbe: C4 Cursor auf Startpos zurück: N... Move: M rechts, runter M +rechts, +runter Cursor relativ zur Cursorpos. bewegen mit + bzw. -
POINT (X, Y)	Gibt den Farbwert des Punktes zurück.
x = POINT (0) y = POINT (1)	Gibt die X- bzw. Y-Koordinate des Cursors zurück.
DIM Feld (400) AS LONG	Speichert einen Bildschirmausschnitt in ein Datenfeld.
GET (X, Y)-(X, Y), Feld	
PUT (10, 10), Feld, PSET	Mit GET abgespeicherte Feld wieder einzeichnen.

Lesen/Schreiben in Datei

OPEN Datei\$ FOR XXXXX AS #1	Beim Dateiöffnen muß eine Nummer (1 - 255) vergeben werden.
... FOR OUTPUT ...	Schreibzugriff. Wenn die Datei bereits vorhanden, dann wird diese gelöscht.
... FOR APPEND ...	Schreibzugriff am Ende einer bereits existierenden Datei. (PRINT / WRITE)
... FOR INPUT ...	Lesezugriff auf Textdatei.
... FOR BINARY ... LEN = 5	Lese- / Schreibzugriff auf Binärdatei. (GET / PUT)
... FOR RANDOM ... LEN = 5	Lese- / Schreibzugriff. Datei ist im Direktzugriffsmodus geöffnet. (GET / PUT) Mit LEN wird die Länge der Datensätze angegeben. (standard: 128 Byt)
CLOSE #1	Datei wieder schließen.
KILL "c:\text.txt"	Löscht eine Datei. (Achtung die Datei ist nun unwiederruflich weg!)
PRINT #1, text\$	Schreibt einen String in eine Textdatei. (FOR OUTPUT / FOR APPEND)
DO UNTIL (EOF (1)) LINE INPUT #1, Zeile\$ LOOP	Zeilenweise aus einer Textdatei (FOR INPUT) lesen . Wenn der Datei-Cursor auf dem Ende der Datei steht, dann gibt der Befehl EOF (=End of File) wahr zurück und die Schleife wird verlassen.
LOF (1)	Gibt die Länge der Datei in Byte an.
PUT #1, Position, Text\$	Schreibt den Text in die Datei, auf angegebene Position. (BINARY/RANDOM) (Deshalb Datensatzlänge beim öffnen von BINARY und RANDOM angeben)
DIM text AS STRING * 3 GET #1, Position, text	Ließt ab Position einen Text aus einer Binärdatei (FOR BINARY) aus. (hier: werden 3 Zeichen ausgelesen)
DIM Bestenliste AS Name GET #1, i, Bestenliste(i)	Ließt ab Position eine Variable aus einer Datei (FOR RANDOM) Hier eignet es sich besonders gut Variablentypen mit TYPE zu erzeugen.
SEEK (1)	Position des Dateizeigers ermitteln. (bei RANDOM in Datensätzen)
SEEK #1, Position	Dateizeiger auf Position setzen.
LOC (1)	Aktuelle Position des Dateizeigers
CHDIR Verzeichnis\$	Ins Verzeichnis wechseln. (Error = bei Fehlschlag)
FILES	Gibt den Inhalt des Aktuellen Verzeichnisses aus.
SHELL DosBefehl\$	führt Dos-Befehle aus: z.B.: "cd " + CHR\$(34) + Verzeichnis + CHR\$(34) "dir *.* /b/ad >liste.tmp"

Sonstiges

GOTO Marke GOSUB Marke	Leider habe ich keine Ahnung wie diese Befhle funktionieren.
SPC (Anzahl)	Überspringt die angegebne Anzahl von Leerzeichen bei PRINT bzw. LPRINT.
NOT Variable	Negiert Bitweise einen Wert (-4 <-> 3 / -3 <-> 2 / -2 <-> 1 / -1 <-> 0)
ON TIMER GOSUB Marke TIMER ON	
ON ERRERO Marke GOSUB ...	
ON PLAY GOSUB ...	