```
' NO TEMP19 VERSION ARRAYS ARE LOADED DIRECTLY
'Comment out shell "erase temp40.dat"

CLS
CLEAR , , 8000     'stack size comes out of the data space so watch out.
CLOSE
'OPEN "ddd.bas" FOR OUTPUT AS #25
ON ERROR GOTO NoTemp40
OPEN "temp40.dat" FOR INPUT AS #40
GOTO readtemp40:

NoTemp40:
RESUME notemp40a
notemp40a:
CLS
LOCATE 12, 1: PRINT "             This module must be run from X-CNTRY"
CLOSE 40
END

readtemp40:
ON ERROR GOTO 0
INPUT #40, desta$, DestIdent$, DestName$, DestCity$, DestState$, DestLat$,
DestLon$, DestElevat$, destvar$, destPatAlt$, destLights$, destfueltype$,
desttower$, destctaf$
CLOSE 40
'SHELL "erase temp40.dat"      'comment out for development

DestLat2$ = ""
FOR x9 = 1 TO 20
temp$ = MID$(DestLat$, x9, 1)
IF temp$ = "." THEN GOTO endgetdestlat
DestLat2$ = DestLat2$ + temp$
NEXT

endgetdestlat:
DestLat$ = DestLat2$
'PRINT #25, "DestLat$"; DestLat$


DestLon2$ = ""
FOR x9 = 1 TO 20
temp$ = MID$(DestLon$, x9, 1)
IF temp$ = "." THEN GOTO endgetdestlon
DestLon2$ = DestLon2$ + temp$
NEXT
endgetdestlon:
DestLon$ = DestLon2$
'PRINT #25, "DestLon$"; DestLon$

takeofftime = INT(TIMER)

temp$ = DATE$
mon$ = MID$(temp$, 1, 2)
day$ = MID$(temp$, 4, 2)
yr$ = MID$(temp$, 9, 2)
temp2$ = TIME$
hr$ = MID$(temp2$, 1, 2)
filenamesim$ = hr$ + mon$ + day$ + yr$ + ".PB"
OPEN filenamesim$ FOR OUTPUT AS #15 LEN = 4000
```

```
tim1 = TIMER
FOR c = 1 TO 999: NEXT
tim2 = TIMER

'PRINT tim1 :PRINT tim2 :PRINT tim2 - tim1
IF tim2 - tim1 > .2 THEN TInterval = 4 ELSE TInterval = 2
'PRINT tinterval :END

CLS
FOR x = 1 TO 4
PRINT
NEXT x
PRINT "               X-CNTRY Copyright 1990, 1991, 1992, 1993"
PRINT "                       by Paul Lamar"
PRINT
PRINT "                                                      "
PRINT
PRINT "                        LAMAR INSTRUMENTS"
PRINT "                        830 Santa Maria St. Suite 303"
PRINT "                        Santa Paula, CA"
PRINT "                        93060 (805) 933 3202"

endit21:
LOCATE 1, 1, 0                              'turn off the cursor

 FOR row1 = 1 TO 18
 FOR ColNum1 = 1 TO 79                       'Get all characters on the line.
 chksum1 = chksum1 + SCREEN(row1, ColNum1) + ColNum1 / row1   'New algol
 NEXT ColNum1
 NEXT row1

'PRINT chksum1                        'Do this when you change the message
'INPUT z$
IF chksum1 <> 61756.53 THEN copyflag = 1
'Any change in the message will screw this up.

IF copyflag <> 1 THEN GOTO goodcopy

'*****************   SMART ASS  CUSTOMER CHANGES
CLS
LOCATE 15, 1
PRINT "              Sorry something is wrong with the program. "
PRINT "              Call Paul Lamar at (805) 933 3202"
END

goodcopy:

maxapli = 60
DIM airportlist(1 TO maxapli) AS STRING * 210
DIM airportprtlst(1 TO maxapli) AS STRING * 210
maxapci = 2 * maxapli

DIM airportcoord(1 TO maxapci)
DIM distogoa(1 TO maxapli)
maxrci = maxapli
DIM rowcol(1 TO maxrci) AS STRING * 18
```

```
ON ERROR GOTO errornosim
OPEN "comunkat.dat" FOR INPUT AS #1
LINE INPUT #1, ctemp$                'this will be used to open the com port
ON ERROR GOTO 0
CLOSE #1

'the comunkat.dat file = this; com1:4800,N,8,1,OP5,RS,RB2048,CD0,DS0,CS0
'More or less. Change for type of GPS
GOTO comunkat1

errornosim:
CLS
LOCATE 10, 1
PRINT "         COMUNKAT.DAT file is missing from this directory."
PRINT "         Please find the file and copy it to this directory."
PRINT "         It should consist of one line and look something like this:"
PRINT "         com1:4800,N,8,1,OP5,RS,RB4096,CD0,DS0,CS0"
END

comunkat1:
OPEN ctemp$ FOR INPUT AS #9 LEN = 8000       'open the com program
'PRINT #25, " Open com point "; TIMER

'GPSS**********  GPS STARTED & SET FLAGS
TIMER ON

ON TIMER(TInterval) GOSUB calculatedata
'Don't be fooled this also turns on the the timer.

GOSUB calculatedata

'PRINT #25, "Start waiting for data "; TIMER

IF gotdataflag > .001 THEN GOTO GotGPSData
LOCATE 20, 1
PRINT "            Searching for data. Please Wait. "
PRINT
PRINT "           Press Esc to end the program."

WFD2:
z = z + 1
IF gotdataflag > .001 THEN GOTO GotGPSData
IF INKEY$ = CHR$(27) THEN GOTO endwfd                          'esc to exit
GOTO WFD2

endwfd:
CLS
CLOSE
SYSTEM

GotGPSData:                                    'if this is activated

'PRINT #25, " got data flag "; gotdataflag; TIMER
'PRINT #25, " waiting for data trys "; z

file1$ = RTRIM$(LTRIM$(LatDeg$))

IF file1$ = "0" THEN file1$ = "00"
```

```
file2$ = RTRIM$(LTRIM$(LonDeg$))

'PRINT #25, " file2$ "; file2$

filename$ = RTRIM$(file1$ + "-" + file2$)


'PRINT #25, " Got the file name "; filename$   '; TIMER
'PRINT #25,

GOSUB getnewfile1

FltStartedFlag = 1
takeofftime$ = TIME$

'FIS*************   FLIGHT INFO SCREEN
FlightInfoScr:
'PRINT #25, "flight info screen "; TIMER
PlotScreenFlag$ = ""
AirportDatScrFlag$ = ""

SCREEN 0
CLS
FltInfoScrFlag$ = "fltinfo"       'best thing I ever did and essential to this
program
TIMER STOP
LOCATE 2, 20: PRINT "You are in Flight Information mode."

'               1         2         3         4         5
'      123456789012345678901234567890123456789012345678901234567890
LOCATE 4, 1: PRINT " Enter data. Cursor keys are active.      "
LOCATE 6, 1: PRINT "                          :";
LOCATE 7, 1: PRINT "                          :";
LOCATE 8, 1: PRINT "                          :";
LOCATE 9, 1: PRINT "                          :";
LOCATE 10, 1: PRINT "    Dest. Designator     :"; DestIdent$;
LOCATE 11, 1: PRINT " Dest. Lat.  deg:min     :"; DestLat$;
LOCATE 12, 1: PRINT " Dest. Lon.  deg:min     :"; DestLon$;
LOCATE 13, 1: PRINT "    Altitude of Dest.    :"; DestElevat$;
LOCATE 14, 1: PRINT "                          :";
LOCATE 15, 1: PRINT " Fuel On Board T.O. Hrs :";
'                    12345678901234567890123456789012345678901234567890
'                              1         2         3
c4 = 38
LOCATE 6, c4: PRINT "    Current pos. Latitude :";
LOCATE 7, c4: PRINT "   Current pos. Longitude :";
LOCATE 8, c4: PRINT "                          :";
LOCATE 9, c4: PRINT "    Ground Speed in Knots :";
LOCATE 10, c4: PRINT "  Distance To Destination :";
LOCATE 11, c4: PRINT "   Heading to Destination :";
LOCATE 12, c4: PRINT "    Estimated Flying Time :";
LOCATE 13, c4: PRINT " Est Time of Arrival Zula :";
LOCATE 14, c4: PRINT "       Greenich Mean Time :";
LOCATE 15, c4: PRINT "       Current Local Time :";

LOCATE 24, 1: PRINT "  Esc to exit.                      F9 Plot Mode. ";
TIMER ON
 ColGetlin = 26
```

```
 colendlin = 36
 VertLimit = 16
 VertPos% = 6                       'Place the cursor at the first line
 HorzPos% = ColGetlin               'Place the cursor at the 20th column
 LeftHorLim% = ColGetlin

'PRINT #25, " timer on point and main info screen loop  "; TIMER

cursorcontrol:                            'Main INPUT SCREEN EDIT loop
 key$ = INKEY$
 LOCATE VertPos%, HorzPos%, 1                        'make a visable cursor.
 IF key$ = CHR$(0) + CHR$(67) AND GotDatFlg$ = "gotdat" AND InfoDatPrtFlg$ =
"InfoPrted" THEN GOTO ThePlotScreen 'F9 key pressed.
 IF key$ = CHR$(27) THEN GOTO endit              'esc to exit
 IF key$ = CHR$(0) + CHR$(72) THEN VertPos% = VertPos% - 1   'down arrow
 IF key$ = CHR$(0) + CHR$(80) THEN VertPos% = VertPos% + 1   'up arrow
 IF key$ = CHR$(0) + CHR$(77) THEN HorzPos% = HorzPos% + 1   'right arrow
 IF key$ = CHR$(0) + CHR$(75) THEN HorzPos% = HorzPos% - 1   'left arrow
 IF key$ = CHR$(8) THEN GOSUB BackSpaceKey               'Backspace key press.
 IF key$ = CHR$(0) + CHR$(83) THEN GOSUB deletekey
 IF key$ = CHR$(13) THEN GOSUB CarrReturn
 IF key$ = CHR$(27) THEN GOTO endit                       'esc to exit
 IF HorzPos% >= colendlin THEN HorzPos% = colendlin
 IF HorzPos% < LeftHorLim% THEN HorzPos% = LeftHorLim%
 IF VertPos% = VertLimit THEN VertPos% = VertLimit - 1'limits cursor movement
 IF VertPos% < 6 THEN VertPos% = 6                    'on the screen
 IF key$ >= CHR$(32) AND key$ <= CHR$(126) THEN GOSUB PrintIt
GOTO cursorcontrol



' ****************** EDITOR  KEY HANDLER ROUTINES
BackSpaceKey:                             'backspaces and erases the character
 TIMER STOP
 HorzPos% = HorzPos% - 1
 IF HorzPos% < LeftHorLim% THEN GOTO BackSpaceKey2
 PRINT " ";                               'erase the character by placing a
 TIMER ON
 RETURN                                   'a space character over it.

BackSpaceKey2:
 TIMER STOP
 BEEP
 HorzPos% = LeftHorLim%
 PRINT " ";
 TIMER ON
 RETURN

CarrReturn:
 TIMER STOP
 HorzPos% = LeftHorLim%
 VertPos% = VertPos% + 1
 TIMER ON
 RETURN

deletekey:
 TIMER STOP
 temp$ = ""                               'Erase the temp string.
 rownum = VertPos%
```

```
 FOR ColNum = (HorzPos% + 1) TO colendlin   'Get all characters on the line.
 charactval = SCREEN(rownum, ColNum)  'Read the screen memory.
 Charact$ = CHR$(charactval)
 temp$ = temp$ + Charact$              'Concatenate all the characters.
 NEXT ColNum
 temp$ = temp$ + " "
LOCATE VertPos%, HorzPos%
PRINT temp$
TIMER ON
RETURN

PrintIt:                               'Legit character so print it on the
 temp$ = ""                            'Erase the temp string. Insert mode
 rownum = VertPos%
TIMER STOP
 FOR ColNum = (HorzPos%) TO colendlin    'Get all characters on the line.
 charactval = SCREEN(rownum, ColNum)     'Read the screen memory.
 Charact$ = CHR$(charactval)
 temp$ = temp$ + Charact$                'Concatenate all the characters.
 NEXT ColNum

LOCATE VertPos%, HorzPos%
PRINT key$
temp$ = MID$(temp$, 1, (colendlin - HorzPos%))
LOCATE VertPos%, (HorzPos% + 1)
PRINT temp$
HorzPos% = HorzPos% + 1               'move the cursor to the next position
 TIMER ON
 RETURN


'*****************************  READ INPUT SCREEN DATA
'hopefully done only once
ReadScreenData:                   'input data was typed into the screen above.
IF FltInfoScrFlag$ <> "fltinfo" THEN GOTO EndReadScreenData
 TIMER STOP

 rownum = 10: NumChar = 8
 GOSUB GetLine
 DestIdent$ = temp$

 rownum = 11: NumChar = 8
 GOSUB GetLine
 DestLat$ = temp$

'PRINT #25, " DestLat$ from scr "; DestLat$

 rownum = 12: NumChar = 8
 GOSUB GetLine
 DestLon$ = temp$

'PRINT #25, " DestLon$ from scr "; DestLon$
'PRINT #25,

 rownum = 13: NumChar = 8
 GOSUB GetLine
 DestAlt$ = temp$
 DestAlt = VAL(DestAlt$)
```

```
 rownum = 15: NumChar = 5
 GOSUB GetLine
 initfuel$ = temp$
 initfuel = VAL(initfuel$)

'XX:XX
DestLat$ = LTRIM$(DestLat$)
DestLatDeg = VAL(MID$(DestLat$, 1, 2))
DestLatDeg2 = DestLatDeg * 60
DestLatMin = VAL(MID$(DestLat$, 4, 2))

DestLat = DestLatMin + DestLatDeg2

'PRINT #25, " DestLat"; DestLat

'0XX:XX

DestLon$ = LTRIM$(RTRIM$(DestLon$))

IF MID$(DestLon$, 1, 1) = "0" THEN DestLon$ = MID$(DestLon$, 2, 5)
IF LEN(DestLon$) = 5 THEN DestLonDeg = VAL(MID$(DestLon$, 1, 2))

IF LEN(DestLon$) = 6 THEN DestLonDeg = VAL(MID$(DestLon$, 1, 3)) 'XXX:XX

DestLonDeg2 = DestLonDeg * 60

IF LEN(DestLon$) = 5 THEN DestLonMin = VAL(MID$(DestLon$, 4, 2))
IF LEN(DestLon$) = 6 THEN DestLonMin = VAL(MID$(DestLon$, 5, 2))

DestLon = DestLonMin + DestLonDeg2

'PRINT #25, " DestLon"; DestLon

TIMER ON
EndReadScreenData:
RETURN

'7************** PRINT DATA ON FLIGHT INFO SCREEN

printGMTonscreen:
TIMER STOP
LOCATE 6, 68: PRINT LatDeg$; : PRINT ":"; : PRINT LatMin$;
LOCATE 7, 67: PRINT LonDeg$; : PRINT ":"; : PRINT lonmin$;

LOCATE 15, 26: PRINT initfuel$;
LOCATE 9, 67: PRINT USING "###.##"; groundspeed; : PRINT "     ";
LOCATE 10, 67: PRINT USING "####.#"; DestDisToGo; : PRINT "    ";
LOCATE 11, 67: PRINT USING "###"; TCourse; : PRINT "   ";

LOCATE 12, 67: PRINT USING "##"; FlyingTime%; : PRINT ":";
LOCATE 12, 70: PRINT USING "##"; FlyingTimeMin%;

LOCATE 13, 67: PRINT USING "##"; EstTimeArr%; : PRINT ":";
LOCATE 13, 70: PRINT USING "##"; EstTimeArrMin%;

LOCATE 14, 67: PRINT GMT$;
LOCATE 15, 67: PRINT TIME$;
TIMER ON
InfoDatPrtFlg$ = "InfoPrted"
```

```
RETURN

'TPS******************** THE PLOT SCREEN -- MAIN LOOP ***************
'with slow note books
'
'backspace key
'delete key
'carraige return
'printit
'last but not leasts the plot screen
'also all the F10 screens

ThePlotScreen:                                  'Set up the plot screen
TIMER OFF
 FltInfoScrFlag$ = ""
 AirportDatScrFlag$ = ""


'PRINT #25, "The plot screen main loop "           '; TIMER

CLS 0
SCREEN 12
WIDTH 80, 60
scale = 7
winsize = 420
corner0 = 5
corner1 = 20                'corner1 = 30    scr 12
corner2 = corner0 + 420     'corner2 = 430   scr 12
corner3 = corner1 + 425     'corner3 = 450   scr 12

VIEW (corner0, corner1)-(corner2, corner3), , 15

'                             1         2         3         4
'                    12345678901234567890123456789012345678901234567890
LOCATE 2, 1: PRINT " ESC TO EXIT       PLOT MODE       F10 AIRPORT DATA";
'                     12345678901234567890123456789012345678901234567890
LOCATE 33, 55: PRINT " USE CURSOR KEYS TO PAN";

PlotScreenFlag$ = "plotscr"
CLS 1

GOSUB plotallairports: 'done each time the lat lon deg change and now
GOSUB SetPlotScrTxt:  'done once each time the screen is set up

IF Eastflag$ = "E" THEN GOSUB moveeast2
IF Westflag$ = "W" THEN GOSUB movewest2
IF Southflag$ = "S" THEN GOSUB moveSouth2
IF Northflag$ = "N" THEN GOSUB moveNorth2

'PRINT #25, "screen 12 plot screen set up "; TIMER

TIMER ON
' ******************  MAIN BACK GROUND LOOP  ************************

Start2:

key$ = INKEY$
IF key$ = CHR$(0) + CHR$(68) THEN GOTO airportdatascr     'F10 key pressed.
```

```
IF key$ = CHR$(0) + CHR$(77) THEN GOSUB moveeast   'right arrow
IF key$ = CHR$(0) + CHR$(75) THEN GOSUB movewest   'left arrow
IF key$ = CHR$(0) + CHR$(80) THEN GOSUB moveSouth   'down arrow
IF key$ = CHR$(0) + CHR$(72) THEN GOSUB moveNorth   'up  arrow
IF key$ = CHR$(27) THEN GOTO endit              'esc to exit
GOTO Start2

'***************** END PLOT SCREEN LOOP ********************


'****************** PANNING SUBROUTINES ******************************
moveNorth:
'PRINT #25, " MOVE NORTH "
newmapflag$ = "newmap"                       'reset by plot all airporst
IF Southflag$ = "S" THEN GOTO NScomeback  'check this this might not work
Northflag$ = "N"
ByPassFlag$ = "BP"
moveNorth2:
'                             1         2         3         4
'                   12345678901234567890123456789012345678901234567890
TIMER STOP
LOCATE 35, 55: PRINT " MAP IS ONE DEGREE NORTH";
LOCATE 37, 55: PRINT "  DOWN ARROW TO RETURN ";
TIMER ON
RETURN

moveSouth:
newmapflag$ = "newmap"                       'reset by plot all airporst
IF Northflag$ = "N" THEN GOTO NScomeback  'check this this might not work
Southflag$ = "S"
ByPassFlag$ = "BP"
moveSouth2:
'                             1         2         3         4
'                   12345678901234567890123456789012345678901234567890
TIMER STOP
LOCATE 35, 55: PRINT " MAP ONE DEGREE SOUTH";
LOCATE 37, 55: PRINT "  UP ARROW TO RETURN   ";
TIMER ON
RETURN

moveeast:
'PRINT #25, " move east "
newmapflag$ = "newmap"                       'reset by plot all airporst
IF Westflag$ = "W" THEN GOTO EWcomeback   'check this this might not work
Eastflag$ = "E"
ByPassFlag$ = "BP"
moveeast2:
'                             1         2         3         4
'                   12345678901234567890123456789012345678901234567890
TIMER STOP
LOCATE 39, 55: PRINT " MAP IS ONE DEGREE EAST";
LOCATE 41, 55: PRINT "  LEFT ARROW TO RETURN ";
TIMER ON
RETURN

movewest:
'PRINT #25, " move west "
newmapflag$ = "newmap"                       'reset by plot all airports
IF Eastflag$ = "E" THEN GOTO EWcomeback   'check this this might not work
```

```
Westflag$ = "W"
ByPassFlag$ = "BP"
movewest2:
'                            1         2         3         4
'                   12345678901234567890123456789012345678901234567890
TIMER STOP
LOCATE 39, 55: PRINT " MAP IS ONE DEGREE WEST";
LOCATE 41, 55: PRINT "  RIGHT ARROW TO RETURN";
TIMER ON
RETURN

EWcomeback:                 'move back to the center
'PRINT #25, " E W come back "
Eastflag$ = ""
Westflag$ = ""
IF Southflag$ = "S" THEN GOTO skipbypassEW
IF Northflag$ = "N" THEN GOTO skipbypassEW
IF Eastflag$ = "E" THEN GOTO skipbypassEW
IF Westflag$ = "W" THEN GOTO skipbypassEW
ByPassFlag$ = ""
skipbypassEW:
'                            1         2         3         4
'                   12345678901234567890123456789012345678901234567890
TIMER STOP
LOCATE 39, 55: PRINT "                         ";
LOCATE 41, 55: PRINT "                         ";
TIMER ON
RETURN

NScomeback:                 'move back to the center
'PRINT #25, " NS come back "
Southflag$ = ""
Northflag$ = ""
IF Southflag$ = "S" THEN GOTO skipbypassFB
IF Northflag$ = "N" THEN GOTO skipbypassFB
IF Eastflag$ = "E" THEN GOTO skipbypassFB
IF Westflag$ = "W" THEN GOTO skipbypassFB
ByPassFlag$ = ""
skipbypassFB:
'                            1         2         3         4
'                   12345678901234567890123456789012345678901234567890
TIMER STOP
LOCATE 35, 55: PRINT "                         ";
LOCATE 37, 55: PRINT "                         ";
TIMER ON
RETURN


'CD********  CACLULATE DATA SUB HAPPENS EVERY TWO SECONDs
calculatedata:    'happens every two seconds
'TIMER OFF        'don't ever put this in! It screws up the background com
'PRINT #25, " Calculate data sub every two seconds "         '; TIMER
'PRINT #25, "FRE(-1) array size available"; FRE(-1)
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

elaptime = INT(TIMER) - takeofftime
ehr = elaptime \ 3600               'integer divide
'sec2 = (elaptime MOD 3600)         'what you get is the remainder with MOD
emin = (elaptime MOD 3600) \ 60     'integer divide the remainder
```

```
esec = (elaptime MOD 3600) MOD 60   'the final remainder is seconds

ehr$ = LTRIM$(STR$(ehr)): IF LEN(ehr$) = 1 THEN ehr$ = "0" + ehr$
emin$ = LTRIM$(STR$(emin)): IF LEN(emin$) = 1 THEN emin$ = "0" + emin$
esec$ = LTRIM$(STR$(esec)): IF LEN(esec$) = 1 THEN esec$ = "0" + esec$

elaptime$ = ehr$ + ":" + emin$ + ":" + esec$
fuel = initfuel - elaptime / 3600
IF PlotScreenFlag$ = "plotscr" THEN GOSUB erasePlane   'must be in front of garmin


'PRINT #25, " Garmin point"        ';TIMER
'PRINT #25, "FRE(-1) array size available"; FRE(-1)
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)


'G*******************    GARMAN
'GARMANroutine:
index1 = 0
GotGPSDataFlag$ = ""

IF LOC(9) < 1 THEN GOTO nodatacomingin   'real important

gotdataflag = gotdataflag + .001
gartemp$ = INPUT$(LOC(9), #9)    'this is the only thing that works
FOR GarX = 1 TO LEN(gartemp$)
gartemp2$ = MID$(gartemp$, GarX, 1)
IF gartemp2$ = "$" THEN GOSUB checkit
IF GotGPSDataFlag$ = "gotgpsdata" THEN GOTO GoodGPSData:
NEXT GarX

GoodGPSData:

gtemp$ = RTRIM$(LTRIM$(gartemp4$))


'the following trick is necessary because INPUT # does not work on
'plain strings.

OPEN "temp45.dat" FOR RANDOM AS #45 'a temp file you use to WRITE & INPUT #
PRINT #45, gtemp$

IF LEN(gtemp$) <> 58 THEN GOTO skipinput45 'you get an error if the $ is wrong

SEEK #45, 1 'use the temp file to parse the string faster than a bunch of MID$
INPUT #45, sentence$, sattime$, Aye$, Lat$, NorS$, Lon$, EorW$, groundspeed$,
heading$
WRITE #15, sattime$, Lat$, NorS$, Lon$, EorW$, groundspeed$, heading$, satdate$
'GPSErrorCnt = GPSErrorCnt + 1

skipinput45:
'IF LEN(gtemp$) <> 58 THEN GOSUB GPSError

GOTO SkipGPSError         'skip over a few following subroutines

'a few local sub routines

'GPSError:
'PRINT #25, "length of gtemp$ "; LEN(gtemp$)
```

```
'PRINT #25, "     GPSErrorCnt "; GPSErrorCnt
'PRINT #25, gtemp$
'WRITE #25, sentence$, sattime$, Aye$, Lat$, NorS$, Lon$, EorW$, groundspeed$,
heading$, satdate$, blank$, blank2$, checksum$
'RETURN

'***************************** CHECKIT SUB CALLED FROM GARMAN ***********
checkit:
'PRINT #25, " check it sub "; TIMER
gartemp3$ = MID$(gartemp$, (GarX + 3), 3)
IF gartemp3$ = "RMC" THEN GOTO RMC
IF gartemp3$ = "GLL" THEN GOTO GLL
'PRINT #25, " end check it sub "; TIMER
'PRINT #25, "FRE(-1) array size available"; FRE(-1)
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)
endcheckit:
GotGPSDataFlag$ = "gotgpsdata"
RETURN

RMC:
gartemp4$ = MID$(gartemp$, GarX, 58)
GOTO endcheckit

GLL:
IF LEN(gartemp$) <> 140 THEN GOTO endcheckit
gartemp5$ = MID$(gartemp$, GarX, 140)

'         1         2         3         4         5         6         7         8
9         10        11        12        13
'12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
345678901234567890123456789012345678901234567890123456789012345678901234567890
'$LCGLL,3016.50,N,08350.63,W  $LCBWC,,3500.00,N,12000.00,W,000,T,001,M,000.0,N,099
$LCAPA,A,A,0.00,R,N,V,A,348,M,099  $LCVTX,001,M,000,N--
'$LCGLL,3045.17,N,08129.38,W  $LCBWC,,3016.51,N,08350.63,W,257,T,260,M,125.1,N,099
$LCAPA,A,A,5.04,L,N,V,A,347,M,099  $LCVTX,115,M,001,N

garlat$ = MID$(gartemp5$, 8, 7) + "0,N,"
garlon$ = MID$(gartemp5$, 18, 8) + "0,W,"
garhead$ = MID$(gartemp5$, 59, 3)
garspeed$ = MID$(gartemp5$, 132, 3)

gartemp4$ = "$GPRMC,221355.756536,A," + garlat$ + garlon$ + "0" + garspeed$ + ".0,"
+ garhead$ + ",030793,,, *20"

'12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
34567890
           '         1         2         3         4         5         6         7
gartemp4$ = MID$(gartemp4$, 1, 58)
GOTO endcheckit
'***************************** END  CHECKIT SUB CALLED FROM GARMAN *******

SkipGPSError:

CLOSE 45

'There are several LON and LAT's;
'The LON/Lat you get when you read the garmin
'LON/LAT you have when you pan which is also used to search
'The LON/LAT of your destination.
```

```
'PRINT #25, "Finished with temp45.dat"           '; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

IF Lat$ = "" THEN GOTO datainvalid
'PRINT #25, " garman lat$ "; Lat$
'XXXX.XXX
LatMin$ = MID$(Lat$, 3, 6)
LatMin = VAL(LatMin$)

'The trouble with lat min is a degree boundry could be right in the
'middle of your approach plate

LatDeg$ = MID$(Lat$, 1, 2)
LatDeg = VAL(LatDeg$)
prelatdeg = LatDeg
'Prelat is used for destination calculations
latcoord = (LatDeg * 60) + LatMin   '=naut miles above the equator
prelat = latcoord
'PRINT #25, " prelat "; prelat

'Lon$ = LTRIM$(Lon$)

'PRINT #25, " garman Lon$ "; Lon$
'PRINT #25,

IF Lon$ = "" THEN GOTO datainvalid
LonDeg$ = MID$(Lon$, 1, 3)

'PRINT #25, " 6 Londeg$ "; Londeg$
'PRINT #25,

lonmin$ = MID$(Lon$, 4, 6)
lonmin = VAL(lonmin$)
LonDeg = VAL(LonDeg$)

loncoord = (LonDeg * 60) + lonmin   'miles west of the meridan
prelon = loncoord

'PRINT #25, " prelon "; prelon

reallatdeg = LatDeg                     'used on print plot screen
reallatdeg$ = LTRIM$(STR$(reallatdeg)) 'used on print plot screen

RealLondeg = LonDeg                       'used on print plot screen
RealLondeg$ = LTRIM$(STR$(RealLondeg))
'PRINT #25, " 8 RealLondeg$ "; RealLondeg$
'PRINT #25,

' 1       2           3  4       5   6       7  8      9   10    11 12 13
'$GPRMC,221355.756536,A,4712.114,N,12213.465,W,0096.5,177,030793,,,*20
'12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012
34567890
'         1         2         3         4         5         6         7

second$ = MID$(sattime$, 5, 2)   'time from the GPS
minute$ = MID$(sattime$, 3, 2)
hour$ = MID$(sattime$, 1, 2)
```

```
'set GMT to atomic clock in satellite.

GMT$ = hour$ + ":" + minute$ + ":" + second$
endgarminsub:

EorW$ = MID$(magvar2$, 3, 1)
magvar2 = VAL(magvar2$)

'*************** CALCULATE DESTINATION DISTANCE TO GO ************

AvgLatDeg = (prelatdeg + DestLatDeg) / 2

'PRINT #25, "AvgLatDeg "; AvgLatDeg

DestCorr = COS(AvgLatDeg / 57.295)    'basic requires radians
                                      'I never thought of it before
                                      'but radiamns might save all that quad
                                      'stuff.

'PRINT #25, " DestCorr "; DestCorr
DifLat = DestLat - prelat                     'all in minutes
IF DifLat = 0 THEN DifLat = .001
DifLon1 = DestLon - prelon                    'all in minutes
DifLon = DifLon1 * DestCorr   'there is a diflon1 because it gets corrected
IF DifLon = 0 THEN DifLon = .001
IF DifLon - DifLat = 0 THEN GOTO SkipThis

'PRINT #25, " DifLon "; DifLon
'PRINT #25, " DifLat "; DifLat

DestDisToGo = ((DifLon ^ 2) + (DifLat ^ 2)) ^ .5
DestDisToGo$ = STR$(DestDisToGo)
'groundspeed$,
groundspeed = VAL(groundspeed$)
IF groundspeed = 0 THEN groundspeed = .001
FlyingTime = DestDisToGo / groundspeed
IF FlyingTime > 99 THEN FlyingTime = 99
FlyingTime% = INT(FlyingTime)                              'hours
FlyingTimeMin% = INT((FlyingTime - FlyingTime%) * 60)   'minutes

'minute$ = MID$(sattime$, 3, 2)
'hour$ = MID$(sattime$, 1, 2)

zulutime = (VAL(minute$) + 60 * VAL(hour$)) / 60   'in decimal hours
EstTimeArr = zulutime + FlyingTime
EstTimeArr% = INT(EstTimeArr)                              'hours
EstTimeArrMin% = INT((EstTimeArr - EstTimeArr%) * 60)   'minutes

SkipThis:

IF DestLon <= prelon AND DestLat >= prelat THEN GOTO FirstQuad
IF DestLon <= prelon AND DestLat <= prelat THEN GOTO SecondQuad
IF DestLon >= prelon AND DestLat <= prelat THEN GOTO ThirdQuad
IF DestLon >= prelon AND DestLat >= prelat THEN GOTO FourthQuad

GOTO endcalc:

FirstQuad:
Opp = (prelon - DestLon) * DestCorr
```

```
Adj = DestLat - prelat
IF Adj = 0 THEN Adj = .001
TCourse = (ATN(Opp / Adj)) * 57.295
GOTO endcalc:

SecondQuad:
Adj = (prelon - DestLon) * DestCorr
IF Adj = 0 THEN Adj = .001
Opp = prelat - DestLat
TCourse = 90 + ((ATN(Opp / Adj)) * 57.295)
GOTO endcalc:

ThirdQuad:
Opp = (DestLon - prelon) * DestCorr
Adj = prelat - DestLat
IF Adj = 0 THEN Adj = .001
TCourse = 180 + ((ATN(Opp / Adj)) * 57.295)
GOTO endcalc:

FourthQuad:
Adj = (DestLon - prelon) * DestCorr
IF Adj = 0 THEN Adj = .001
Opp = DestLat - prelat
TCourse = 270 + ((ATN(Opp / Adj)) * 57.295)

endcalc:

TCourse$ = STR$(TCourse)
BestAlt = ((DestDisToGo * 6080) * SIN(3 / 57.295)) + DestAlt
IF BestAlt > 20000 THEN BestAlt = 20000

IF EorW$ = "E" THEN magheading = heading - magvar2 ELSE magheading = heading +
magvar

IF magheading > 360 THEN magheading = magheading - 360
magheading$ = LTRIM$(STR$(magheading))

heading = VAL(heading$)
Kts = VAL(groundspeed$)
mph = Kts * 1.1515

'************** NOW SEE IF THE MAP CHANGED AND PLOT A NEW SCREEN ****

IF Northflag$ <> "N" THEN GOTO SkipNorth
LatDeg = INT(LatDeg + 1)               'it does so do it:
LatDeg$ = LTRIM$(RTRIM$(STR$(LatDeg)))

SkipNorth:
IF Southflag$ <> "S" THEN GOTO SkipSouth
LatDeg = INT(LatDeg - 1)
LatDeg$ = LTRIM$(RTRIM$(STR$(LatDeg)))

SkipSouth:
latdegplus1$ = LTRIM$(STR$(LatDeg + 1))    'used on print plot screen

'PRINT #25, Eastflag$;

IF Eastflag$ <> "E" THEN GOTO SkipLonEast
LonDeg = INT(LonDeg - 1)
```

```
LonDeg$ = LTRIM$(RTRIM$(STR$(LonDeg)))

SkipLonEast:

IF Westflag$ <> "W" THEN GOTO SkipLonWest
LonDeg = INT(LonDeg + 1)
LonDeg$ = LTRIM$(RTRIM$(STR$(LonDeg)))

'PRINT #25, Westflag$;

SkipLonWest:

LonDegPrt$ = LTRIM$(STR$(LonDeg + 1))   'used on left side of plot screen

'PRINT #25, Westflag$;

IF firsttimeflag$ = "firsttime" THEN GOTO skipfirsttime
'PRINT #25, " NO SKIP FIRST TIME  oldlatdeg = LatDeg   &   oldlondeg = LonDeg   "
 oldlatdeg = LatDeg
 oldlondeg = LonDeg
firsttimeflag$ = "firsttime"

skipfirsttime:

'PRINT #25, " SKIP FIRST TIME"


IF LatDeg < oldlatdeg THEN GOSUB getnextsouth  '& a whole bunch of other stuff
IF LatDeg > oldlatdeg THEN GOSUB getnextnorth
IF LonDeg < oldlondeg THEN GOSUB getnexteast
IF LonDeg > oldlondeg THEN GOSUB getnextwest

'when you come back here you might come back with a whole new map

IF FltInfoScrFlag$ = "fltinfo" THEN GOSUB ReadScreenData
IF FltInfoScrFlag$ = "fltinfo" THEN GOSUB printGMTonscreen

IF PlotScreenFlag$ = "plotscr" THEN GOSUB plotplane   'plot plane every 2 sec
IF PlotScreenFlag$ = "plotscr" THEN GOSUB PrtPlotScrData 'update every 2 sec

'PRINT #25, " End the calculate data sub "             '; TIMER
'PRINT #25, "FRE(-1) array size available"; FRE(-1)
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

'TIMER OFF        'Don't ever put this in. It screws up the back ground com
endallcalc:
'TIMER ON         'Don't ever put this in. It screws up the back ground com
RETURN            'go back to back ground


'GNS*********************  GET NEXT SOUTH  *********************
getnextsouth:               'heading south
oldlatdeg = LatDeg
GOTO getnewfile1          'continue the sub else where

getnextnorth:                   'heading north latpix is decreasing
'PRINT #25, " getnextnorth: "
oldlatdeg = LatDeg
GOTO getnewfile1
```

```
getnexteast:
oldlondeg = LonDeg
GOTO getnewfile1

getnextwest:
oldlondeg = LonDeg
GOTO getnewfile1

'GNF1*********************   GET NEW FILE  ONE
getnewfile1:

'PRINT #25, " Get new file sub "             ';TIMER
'PRINT #25, "FRE(-1) array size available"; FRE(-1)
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

'CLS 1

'TBS************************* THE BIG SEARCH ********************

IF LatDeg$ = "0" THEN LatDeg$ = "00"
IF LEN(LonDeg$) = 2 THEN LonDeg$ = "0" + LonDeg$
'corrects for adding one to to 083 longitude.
'It is different with LonDeg" = "083" verses LonDeg$ = "83"
'           30      083 or 83 makes a big difference

test = VAL(LatDeg$ + LonDeg$)

' "test" is the number you are searching for
'it is the first place in the file where you find the first occurance
'off the one degree block of data you are looking for.


FOR apli = 1 TO maxapli    'clear the airport list & prt list arrays for F10
airportlist(apli) = ""
airportprtlst(apli) = ""
NEXT apli
apli = 0               'set the array index to 0   pre indexed later

FOR apci = 1 TO maxapci     'clear the airport coord array  for plotting
airportcoord(apci) = 0
NEXT apci
apci = 0               'set the airport coord array index to 0

scale = 7             'this is done when the plot screen is set up
winsize = 420         'it is done here again so the first time through works!

FOR rci = 1 TO maxrci     'clear the airport rowcol array
rowcol(rci) = ""
NEXT rci

rci = 0               'set the airport rowcol array index to 0
endrci = 0

oldprownum% = 0
oldpcolnum% = 0

OPEN "airport.dat" FOR INPUT AS #51 LEN = 24000
LINE INPUT #51, templen$
```

```
reclen = LEN(templen$) + 2
SEEK #51, 1
filelen = 2126615                'this is critical
numrecords = INT(filelen / reclen)  '10795 this is critical don't add to the length
low = 0
high = numrecords + 1
mid = INT((low + high) / 2)      'low starts at 0   mid starts at count/2
mid2 = mid * reclen

'tim1# = TIMER

ON ERROR GOTO errorbigsearch
start:
IF mid2 = oldmid2 THEN GOTO nomatchfound
oldmid2 = mid2

SEEK #51, mid2

INPUT #51, f1$, f2$, f3$, f4$, f5$, f6$, f7$, f8$, f9$, f10$, f11$, f12$, f13$,
f14$, f15$

testtemp = VAL(MID$(f6$, 1, 2) + MID$(f7$, 1, 3))

'PRINT #25, "     test "; test
'PRINT #25, "test temp "; testtemp
'PRINT #25, "      mid "; mid
'PRINT #25, "     mid2 "; mid2

' 1    2         3                                          4
5       6              7          8       9     10     11       12            13
14    15
'"A","04L  ","CONOVER AIR LODGE                          ","FRAZIER PARK
","CA","34:45.665N","119:03.539W"," 5160"," 3000","14E","    "," "," "
","  ","       "
'"A","SZP  ","SANTA PAULA                                ","SANTA PAULA
","CA","34:20.833N","119:03.672W","  245"," 2650","14E","600 "," "," 100LL80
","  ","122.900"
'"A","CMA  ","CAMARILLO                                  ","CAMARILLO
","CA","34:12.825N","119:05.659W","   75"," 6010","14E","800 ","DUSK-DAWN","A
100LL80        ","Y","128.200"
'"A","NTD  ","POINT MUGU NAWS                            ","POINT MUGU
","CA","34:07.217N","119:07.256W","   12","11100","14E","    ","DUSK-DAWN","
","Y","       "
'"A","OXR  ","OXNARD                                     ","OXNARD
","CA","34:12.048N","119:12.432W","   43"," 5950","14E","1000","DUSK-DAWN","A
100LL          ","Y","120.000"
'"A","CL25 ","7R RANCH                                   ","CUYAMA
","CA","34:56.131N","119:27.107W"," 3176"," 2350","14E","    "," "," "
","  ","       "
'"A","3CA3 ","DIXON                                      ","CUYAMA
","CA","34:55.331N","119:31.557W"," 2440"," 2590","14E","    "," "," "
","  ","       "
'"A","L88  ","NEW CUYAMA                                 ","NEW CUYAMA
","CA","34:56.248N","119:41.307W"," 2203"," 3970","14E","1000","  "," "
","  ","122.900"
'"A","SBA  ","SANTA BARBARA MUNI                         ","SANTA BARBARA
","CA","34:25.573N","119:50.421W","   10"," 6049","14E","    ","DUSK-DAWN","A
100LL          ","Y","119.700"
```

```
'"A","CA97 ","CHRISTY AIRSTRIP                          ","SANTA BARBARA
","CA","34:01.183N","119:51.174W","  250"," 2507","14E","    ","         ","
","  ","       "
'"A","SZN  ","SANTA CRUZ ISLAND                         ","SANTA BARBARA
","CA","34:03.633N","119:54.907W","   50"," 2150","14E","    ","         ","
","  ","       "


IF testtemp = test THEN GOTO printit51

IF testtemp > test THEN            'lowerit
'         ISGT
high = mid                          'mid will get lower
mid = INT((low + high) / 2)
mid2 = mid * reclen
END IF

IF testtemp < test THEN              ' raiseit
'          ISLT
low = mid
mid = INT((low + high) / 2)         'mid will be higher
mid2 = mid * reclen                  'INT after the divide is crucial
END IF

GOTO start

'Instead of doing the following kludge we should see if we can load
'the arrays directly

printit51:

'PRINT #25, LatDeg$; ";LatDeg$;"; LonDeg$; ";LonDeg$"

index51 = mid2 - (60 * reclen)            'go back to insure you get all

IF mid2 <= (60 * reclen) THEN index51 = 1  'neccessary to get the first ten

SEEK #51, index51

FOR x51 = 1 TO 120               'get 120 airports in total

INPUT #51, f1$, f2$, f3$, f4$, f5$, f6$, f7$, f8$, f9$, f10$, f11$, f12$, f13$,
f14$, f15$

'WRITE #25, f2$, f6$, f7$

newtest2$ = MID$(f7$, 1, 3)
newtest3$ = MID$(f6$, 1, 2)

'PRINT #25, "    LonDeg$ "; LonDeg$; "   "; " newtest2$ "; newtest2$
'PRINT #25, "    LatDeg$ "; LatDeg$; "   "; " newtest3$ "; newtest3$

IF newtest2$ = LonDeg$ AND newtest3$ = LatDeg$ THEN GOSUB loadarraydirect

NEXT x51


endprt51:
CLOSE #51
```

```
'tim2# = TIMER
'PRINT #25, " Search routine time "; tim2# - tim1#
'PRINT #25, " number of lines added to the arrays "; cntarr
'PRINT #25, " Return from search "           '; TIMER
ON ERROR GOTO 0
GotDatFlg$ = "gotdat"
IF PlotScreenFlag$ = "plotscr" THEN GOSUB plotallairports:

RETURN


'LTA************************* LOAD THE AIRPORT  ARRAYS  SUB

loadarraydirect:     'This is called once for each line of data
'cntarr = cntarr + 1

'PRINT #25, " Load the airport arrays  DIRECT"                 '; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)
'PRINT #25, "Max Array index "; maxapli

magvar2$ = f10$
' 1    2         3                                          4
5       6            7          8       9      10    11      12               13
14   15
'"A","01K  ","CALDWELL MUNI                              ","CALDWELL
","KS","37:02.168","097:35.186"," 1157"," 2380","07E","    ","DUSK-DAWN","
","  ","122.900"
'"A","02CO ","MC CULLOUGH                                ","MONTE VISTA
","CO","37:38.600","106:02.818"," 7615"," 2670","12E","    ","         ","
","  ","       "
'"A","03MO ","CAHOOCHIE                                  ","URBANA
","MO","37:53.068","093:07.894"," 1010"," 2200","04E","    ","         ","
","  ","       "
'"A","03V  ","SAN LUIS                                   ","SAN LUIS
","CO","37:11.500","105:27.784"," 7969"," 4300","11E","    ","         ","
","  ","122.900"

f1$ = LTRIM$(f1$)
IF f1$ = "L" THEN GOTO skip2:
IF f1$ = "C" THEN GOTO skip2
IF f1$ = "O" THEN GOTO skip2

apli = apli + 1

'PRINT #25, " Array index apli "; apli

airportlist(apli) = fd2$ + f3$ + f4$ + f5$ + f8$ + f9$ + f10$ + f11$ + f12$ + f13$
+ f14$

'PRINT #25, "apli ="; apli; " "; airportlist(apli)
'PRINT #25, "MaxPrtCnt "; MaxPrtCnt

IF MID$(f14$, 1, 1) = " " THEN f14$ = "No"

IF MID$(f13$, 1, 1) = " " THEN f13$ = " None             "
IF MID$(f12$, 1, 1) = " " THEN f12$ = " None    "


'         desig NAME      City          State        Elev.       Rwy Len.
```

```
temp45$ = f2$ + f3$ + MID$(f4$, 1, 23) + f5$ + " " + f8$ + "   " + f9$ + "     "

'         Var.         Lighting Fuel types avail. Tower           CTAF
temp46$ = f10$ + "  " + f12$ + " " + f13$ + " " + f14$ + "    " + f15$

airportprtlst(apli) = temp45$ + temp46$

skip2:

MaxPrtCnt = apli


'LACA*****************    LOAD AIRPORT COORDINATE ARRAYS FOR PLOTTING

'loadaparrays:        'This is called once for each line of
'PRINT #25, " load airport coordinate arrays:"             '; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

'PRINT #25, f2$; " "; f6$; " "; f7$

 Ident$ = RTRIM$(LTRIM$(f2$))
 latapdeg = VAL(LTRIM$(MID$(f6$, 1, 2)))
 latapmin = VAL(MID$(f6$, 4, 6))
 AbsLatAP = latapdeg * 60 + latapmin
 apci = apci + 1
 airportcoord(apci) = AbsLatAP                      'for distances calc

'PRINT #25, " airportcoord(apci) "; airportcoord(apci)

'PRINT #25, " winsize "; winsize;
'PRINT #25, "       scale "; scale

 PlotAirLat3 = winsize - (latapmin * scale)   'from above latapmin

 prownum% = INT(PlotAirLat3 / 8)

prownum2% = prownum%
 IF prownum% < 4 THEN prownum2% = 4         'ELSE
 IF prownum% >= 55 THEN prownum2% = 55         'ELSE prownum2% = prownum%

 lonapdeg = VAL(MID$(f7$, 1, 3))
 lonapmin = VAL(MID$(f7$, 5, 6))
 AbsLonAP = lonapdeg * 60 + lonapmin
 apci = apci + 1
 airportcoord(apci) = AbsLonAP                       'for distances calc

'PRINT #25, " airportcoord(apci) "; airportcoord(apci)

'*********************************************************************
 PlotAirLong3 = winsize - (lonapmin * scale)       'pixels reversed
 pcolnum% = INT(PlotAirLong3 / 8)                   '640/80 = 8

 pcolnum% = (pcolnum% + 1)                'center on 2nd character

pcolnum2% = pcolnum%

 IF pcolnum% <= 2 THEN pcolnum2% = 2
 IF pcolnum% >= 50 THEN pcolnum2% = 50
```

```
lenident = LEN(Ident$)
Ident$ = Ident$ + SPACE$(6 - lenident)

pcolnum2$ = LTRIM$(RTRIM$(STR$(pcolnum2%)))
lenpc2 = LEN(pcolnum2$)
pcolnum3$ = pcolnum2$ + SPACE$(6 - lenpc2)
prownum2$ = LTRIM$(RTRIM$(STR$(prownum2%)))
lenpr2 = LEN(prownum2$)
prownum3$ = prownum2$ + SPACE$(6 - lenpr2)
rci = rci + 1
finalrc$ = Ident$ + prownum3$ + pcolnum3$
rowcol(rci) = finalrc$
'PRINT #25, " rowcol(rci) "; rowcol(rci)
'PRINT #25,
endrci = rci

RETURN



'PAA******************* PLOT ALL AIRPORTS  ON PLOT SCREEN **************
'done only when map changes and at the beginning.
'something calls this at the beginning

plotallairports:

'PRINT #25, " Plot all airports  ";        '; TIMER

'PRINT #25, "FRE(-2) stack size available"; FRE(-2)
'PRINT #25, "  PlotScreenFlag$ "; PlotScreenFlag$
'PRINT #25,

IF PlotScreenFlag$ <> "plotscr" THEN GOTO endplotallairports
CLS 1

rci3 = 0
prtrowcol25:
rci3 = rci3 + 1: IF rci3 > endrci THEN GOTO endprtrowcol25
temp25$ = rowcol(rci3):                                    'PRINT #25, temp25$;
rci3 = rci3 + 1: IF rci3 > endrci THEN GOTO endprtrowcol25
temp25$ = rowcol(rci3):                                    'PRINT #25, temp25$;
rci3 = rci3 + 1: IF rci3 > endrci THEN GOTO endprtrowcol25
temp25$ = rowcol(rci3):                                    'PRINT #25, temp25$
GOTO prtrowcol25
endprtrowcol25:

'CRCA************* CHECK ROW COL ARRAY ******************

'PRINT #25, " check row col for coincedenses"       '; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

'check the array

FOR rci = 1 TO endrci
temp$ = rowcol(rci)
'PRINT #25, temp$

FOR rci98 = 1 TO endrci
temp2$ = rowcol(rci98)
```

```
IF temp$ <> temp2$ THEN GOSUB checkpos98
NEXT

'PRINT #25, rowcol(rci)
NEXT

'PRINT #25, " End check row col for coincedenses"; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

FOR rci = 1 TO endrci
temp$ = rowcol(rci)
Ident$ = MID$(temp$, 1, 6)
tmprow = VAL(MID$(temp$, 7, 6))
tmpcol = VAL(MID$(temp$, 13, 6))
LOCATE tmprow, tmpcol:  PRINT RTRIM$(LTRIM$(Ident$));
NEXT rci

newmapflag$ = ""

endplotallairports:

'PRINT #25, "End plot all airports "; TIMER
'PRINT #25, " endrci "; endrci
'PRINT #25, "FRE(-1) array size available"; FRE(-1)
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

RETURN

checkpos98:
IF MID$(temp$, 7, 6) = MID$(temp2$, 7, 6) THEN GOSUB samerow
RETURN

samerow:
valcol1 = VAL(MID$(temp$, 13, 6))
valcol2 = VAL(MID$(temp2$, 13, 6))
IF valcol1 = valcol2 THEN GOSUB samepos
IF valcol1 = valcol2 + 1 THEN GOSUB samepos
IF valcol1 = valcol2 + 2 THEN GOSUB samepos
IF valcol1 = valcol2 + 3 THEN GOSUB samepos
IF valcol1 = valcol2 + 4 THEN GOSUB samepos
RETURN


samepos:
newrowval$ = RTRIM$(LTRIM$(STR$(VAL(MID$(temp$, 7, 6)) + 1)))
lennewrowval = LEN(newrowval$)
newrowval2$ = newrowval$ + SPACE$(6 - lennewrowval)
rowcol(rci) = MID$(temp$, 1, 6) + newrowval2$ + MID$(temp2$, 13, 6)
RETURN


'PADS****** PRINT AIRPORT DATA SCREEN  F10 PRESSED FIRST PAGE ******
airportdatascr:
TIMER STOP   'if you do this I think the comm buffer overflows
'PRINT #25, " print airport data screen. F10 pressed "       '; TIMER

GOSUB distancetogo

PlotScreenFlag$ = ""
```

```
FltInfoScrFlag$ = ""
AirportDatScrFlag$ = "aptdatscr"

CLS 0
SCREEN 12
WIDTH 80, 60
lowerlimit = 1
upperlimit = 14
GOSUB getairdatascr
'TIMER STOP
LOCATE 59, 1: PRINT "  First Page   Page Down   Enter for Approach (Cursor Key
Select)  F9 Plot Mode";

VertPos% = 2
LOCATE VertPos%, 1: PRINT "=====";
'TIMER ON

start4:
TIMER STOP
 key$ = INKEY$
IF key$ = CHR$(0) + CHR$(72) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% - 4
   IF VertPos% < 2 THEN VertPos% = 2                    'on the screen
   LOCATE VertPos%, 1: PRINT "=====";


END IF

IF key$ = CHR$(0) + CHR$(80) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% + 4
   IF VertPos% >= 54 THEN VertPos% = 54      'limits cursor movement
   LOCATE VertPos%, 1: PRINT "=====";
END IF
TIMER ON

 IF key$ = CHR$(0) + CHR$(67) THEN GOTO EndAllPrtAirScr:   'F9 key pressed.
 IF key$ = CHR$(13) THEN GOTO RunApproach:                 'Enter  key pressed.
 IF key$ = CHR$(0) + CHR$(81) THEN GOTO secondpage:        'Page Down
 IF key$ = CHR$(0) + CHR$(73) THEN GOTO EndAllPrtAirScr:   'page Up
GOTO start4

'***************** RUN APPROACH *************************
RunApproach:
 TIMER STOP
  temp$ = ""
  FOR x = 6 TO 11
  tempchr$ = CHR$(SCREEN((VertPos% - 1), x))
  temp$ = temp$ + tempchr$
  NEXT x

OPEN "transfer.dat" FOR OUTPUT AS #30
PRINT #30, temp$
'PRINT #30, "yes"
CLOSE 30

SCREEN 0
CLEAR
```

```
SHELL "MODULE3"
END


'SCREEN 0

'GOTO goodcopy:


EndAllPrtAirScr:
'PRINT #25, " End print on airport data screen routine "; TIMER
GOTO ThePlotScreen:

getairdatascr:
FOR indcnt = lowerlimit TO upperlimit      'indcnt is index to airport list
IF indcnt > MaxPrtCnt THEN GOTO endcount
temp$ = airportprtlst(indcnt)            'one long line of data
dtgtemp = distogoa(indcnt)               'remember! arrays are random access
PRINT USING ("###.#"); dtgtemp;
PRINT " ";
PRINT MID$(temp$, 1, 74)
PRINT "      Elev   Rwy Len  Var  Lighting  Fuel types         Tower  CTAF "
PRINT "      ";
PRINT MID$(temp$, 76, 65)
PRINT
NEXT indcnt
endcount:
RETURN

'*************** PRINT ON AIRPORT DATA SCREEN  SECOND PAGE ***************
secondpage:
CLS 0
lowerlimit = 15
upperlimit = 28
GOSUB getairdatascr

TIMER STOP
LOCATE 59, 1: PRINT "  Second Page   Page Down   Enter for Approach (Cursor Key
Select)  F9 Plot Mode";

VertPos% = 2
LOCATE VertPos%, 1: PRINT "=====";
start42:
 key$ = INKEY$

IF key$ = CHR$(0) + CHR$(72) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% - 4
   IF VertPos% < 2 THEN VertPos% = 2                    'on the screen
   LOCATE VertPos%, 1: PRINT "=====";


END IF

IF key$ = CHR$(0) + CHR$(80) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% + 4
   IF VertPos% >= 54 THEN VertPos% = 54      'limits cursor movement
```

```
   LOCATE VertPos%, 1: PRINT "=====";
END IF
TIMER ON

 IF key$ = CHR$(0) + CHR$(67) THEN GOTO EndAllPrtAirScr:      'F9 key pressed.
 IF key$ = CHR$(13) THEN GOTO RunApproach:                  'Enter  key pressed.
 IF key$ = CHR$(0) + CHR$(73) THEN GOTO airportdatascr     'page Up
 IF key$ = CHR$(0) + CHR$(81) THEN GOTO ThirdPage          'Page down

GOTO start42


'*************** PRINT ON AIRPORT DATA SCREEN  THIRD PAGE ***************
ThirdPage:
CLS 0
lowerlimit = 29
upperlimit = 42
GOSUB getairdatascr

TIMER STOP
LOCATE 59, 1: PRINT "  Third Page   Page Down   Enter for Approach (Cursor Key
Select)  F9 Plot Mode";

VertPos% = 2
LOCATE VertPos%, 1: PRINT "=====";
start43:
 key$ = INKEY$

IF key$ = CHR$(0) + CHR$(72) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% - 4
   IF VertPos% < 2 THEN VertPos% = 2                    'on the screen
   LOCATE VertPos%, 1: PRINT "=====";


END IF

IF key$ = CHR$(0) + CHR$(80) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% + 4
   IF VertPos% >= 54 THEN VertPos% = 54      'limits cursor movement
   LOCATE VertPos%, 1: PRINT "=====";
END IF
TIMER ON

 IF key$ = CHR$(0) + CHR$(67) THEN GOTO EndAllPrtAirScr:      'F9 key pressed.
 IF key$ = CHR$(13) THEN GOTO RunApproach:                  'Enter key pressed.
 IF key$ = CHR$(0) + CHR$(81) THEN GOTO fourthpage         'Page down
 IF key$ = CHR$(0) + CHR$(73) THEN GOTO secondpage         'page Up

GOTO start43


'*************** PRINT ON AIRPORT DATA SCREEN  FOURTH PAGE ***************
fourthpage:
CLS 0
lowerlimit = 43
upperlimit = 56
GOSUB getairdatascr
```

```
TIMER STOP
LOCATE 59, 1: PRINT "  Fourth Page   Page Down   Enter for Approach (Cursor Key
Select)  F9 Plot Mode";

VertPos% = 2
LOCATE VertPos%, 1: PRINT "=====";
start44:
 key$ = INKEY$

IF key$ = CHR$(0) + CHR$(72) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% - 4
   IF VertPos% < 2 THEN VertPos% = 2                    'on the screen
   LOCATE VertPos%, 1: PRINT "=====";


END IF

IF key$ = CHR$(0) + CHR$(80) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% + 4
   IF VertPos% >= 54 THEN VertPos% = 54      'limits cursor movement
   LOCATE VertPos%, 1: PRINT "=====";
END IF
TIMER ON
 IF key$ = CHR$(0) + CHR$(67) THEN GOTO EndAllPrtAirScr:      'F9 key pressed.
 IF key$ = CHR$(13) THEN GOTO RunApproach:                  'Enter key pressed.
 IF key$ = CHR$(0) + CHR$(73) THEN GOTO ThirdPage     'page Up
 IF key$ = CHR$(0) + CHR$(81) THEN GOTO fifthpage     'Page down

GOTO start44

'*************** PRINT ON AIRPORT DATA SCREEN  FIFTH PAGE ***************
fifthpage:
CLS 0
lowerlimit = 57
upperlimit = 70
GOSUB getairdatascr
TIMER STOP
LOCATE 59, 1: PRINT "  Fifth Page   Page Up   Enter for Approach (Cursor Key
Select)  F9 Plot Mode";

VertPos% = 2
LOCATE VertPos%, 1: PRINT "=====";
start45:
 key$ = INKEY$

IF key$ = CHR$(0) + CHR$(72) THEN
   LOCATE VertPos%, 1: PRINT "     ";
   VertPos% = VertPos% - 4
   IF VertPos% < 2 THEN VertPos% = 2                    'on the screen
   LOCATE VertPos%, 1: PRINT "=====";


END IF

IF key$ = CHR$(0) + CHR$(80) THEN
   LOCATE VertPos%, 1: PRINT "     ";
```

```
   VertPos% = VertPos% + 4
   IF VertPos% >= 54 THEN VertPos% = 54      'limits cursor movement
   LOCATE VertPos%, 1: PRINT "=====";
END IF
TIMER ON
 IF key$ = CHR$(0) + CHR$(67) THEN GOTO EndAllPrtAirScr:      'F9 key pressed.
 IF key$ = CHR$(13) THEN GOTO RunApproach:                 'Enter key pressed.
 IF key$ = CHR$(0) + CHR$(73) THEN GOTO fourthpage     'page Up

GOTO start45

'********************** GET LINE  FROM THE SCREEN
GetLine:
 temp$ = ""                              'Erase the temp string.
 FOR ColNum = ColGetlin TO 24 + NumChar       'Get all characters on the line.
 charactval = SCREEN(rownum, ColNum)   'Read the screen memory.
 Charact$ = CHR$(charactval)
 temp$ = temp$ + Charact$              'Concatenate all the characters.
 NEXT ColNum
 temp$ = LTRIM$(RTRIM$(temp$))         'Trim trailing right blanks
 RETURN                                'Go back with the data in the Temp$

'PPSD******************** SET UP  PLOT SCREEN
SetPlotScrTxt: :  'used in plot mode only
IF PlotScreenFlag$ <> "plotscr" THEN GOTO EndSetPlotScrTxt:
TIMER STOP
'PRINT #25, " Set up plot screen "; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

cn2 = 56
LOCATE 7, cn2: PRINT " LAT POSITION  ";
LOCATE 9, cn2: PRINT " LON POSITION  ";
LOCATE 11, cn2: PRINT "    LOCAL TIME   ";
LOCATE 13, cn2: PRINT "GREENWICH TIME   ";
LOCATE 15, cn2: PRINT "  ELAPSED TIME   ";
LOCATE 17, cn2: PRINT "                 ";
LOCATE 19, cn2: PRINT "REMAINING FUEL   ";
LOCATE 21, cn2: PRINT "MAGNETIC HEADING  ";
LOCATE 23, cn2: PRINT "MAGNETIC VARIAT   ";
LOCATE 25, cn2: PRINT "TRUE HEADING      ";
LOCATE 27, cn2: PRINT "GROUND SPEED KTS  ";
LOCATE 29, cn2: PRINT "GROUND SPEED MPH  ";
LOCATE 44, cn2: PRINT "  COURSE TO DEST  ";
LOCATE 46, cn2: PRINT "DISTANCE TO DEST  ";
LOCATE 48, cn2: PRINT "BEST DESCENT ALT  ";
LOCATE 50, cn2: PRINT "FLYING TIME DEST  ";
LOCATE 52, cn2: PRINT "EST TIME ARR ZULU ";

'PRINT #25, " End set up plot screen "; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)
TIMER ON
EndSetPlotScrTxt: :
RETURN

'PPS******************  PRINT PLOT SCREEN DATA
PrtPlotScrData: :               'used in plot mode only
IF PlotScreenFlag$ <> "plotscr" THEN GOTO EndPrtPlotScr
TIMER STOP
'PRINT #25, "Print the plot screen data "; TIMER
```

```
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

cn1 = 71

LOCATE 3, 55: PRINT latdegplus1$; ":00"

LOCATE 7, cn1: PRINT "        ";       'erase
LOCATE 7, cn1: PRINT reallatdeg$;
PRINT ":";
PRINT LatMin$;

LOCATE 9, cn1: PRINT "        ";      'erase
LOCATE 9, (cn1 - 1): PRINT RealLondeg$;
PRINT ":";
PRINT lonmin$;
LOCATE 11, (cn1 + 1): PRINT TIME$;
LOCATE 13, (cn1 + 1): PRINT GMT$;
LOCATE 15, (cn1 + 1): PRINT elaptime$;
LOCATE 17, cn1: PRINT ;
LOCATE 19, cn1: PRINT USING "  ##.##"; fuel;
LOCATE 21, (cn1 + 2): PRINT "       ";
LOCATE 21, (cn1 + 3): PRINT magheading$
LOCATE 23, (cn1 + 2): PRINT "       ";
LOCATE 23, (cn1 + 3): PRINT magvar2$;
LOCATE 25, (cn1 + 2): PRINT "       ";
LOCATE 25, (cn1 + 3): PRINT heading$;
LOCATE 27, (cn1 + 2): PRINT "       ";
LOCATE 27, (cn1 + 3): PRINT USING "###.#"; Kts;
LOCATE 29, (cn1 + 2): PRINT "       ";
LOCATE 29, (cn1 + 3): PRINT USING "###.#"; mph;
LOCATE 44, (cn1 + 4): PRINT "    ";
LOCATE 44, (cn1 + 4): PRINT USING "###"; TCourse;

LOCATE 46, (cn1 + 3): PRINT "     ";
LOCATE 46, (cn1 + 3): PRINT USING "####"; DestDisToGo;

LOCATE 48, (cn1 + 3): PRINT "     ";
LOCATE 48, (cn1 + 3): PRINT USING "#####"; BestAlt;

LOCATE 50, (cn1 + 3): PRINT "       ";



LOCATE 50, (cn1 + 3): PRINT USING "##:"; FlyingTime%;
LOCATE 50, (cn1 + 6): PRINT USING "##"; FlyingTimeMin%;

LOCATE 52, (cn1 + 3): PRINT "       ";
LOCATE 52, (cn1 + 3): PRINT USING "##:"; EstTimeArr%;    'hours
LOCATE 52, (cn1 + 6): PRINT USING "##"; EstTimeArrMin%;  'minutes


LOCATE 56, 55: PRINT LatDeg$; ":00"
LOCATE 57, 2: PRINT LonDegPrt$; ":00                                                    ";
LonDeg$; ":00"
'PRINT #25, " End print the plot screen data "; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)
TIMER ON
EndPrtPlotScr:
RETURN
```

```
'CDTG*****************   CALCULATE  DISTANCES TO GO

distancetogo:
'TIMER STOP
'PRINT #25, " Calculate distances to go. "; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)

ON ERROR GOTO edistancetogo       'exit mechanizam

apci = 0
dtgi1 = 0


'PRINT #25, " latcoord "; latcoord;
'PRINT #25, " loncoord "; loncoord

startdtg:
apci = apci + 1
'PRINT #25, " Apci "; apci

'PRINT #25, " ap lat coord "; airportcoord(apci);

APAvgLatDeg = ((latcoord + airportcoord(apci)) / 2) \ 60

'PRINT #25, "APAvgLatDeg"; APAvgLatDeg

AirPortCorr = COS(APAvgLatDeg / 57.295)    'basic requires radians

'PRINT #25, " AirPortCorr "; AirPortCorr

ydif = latcoord - airportcoord(apci)  'this will be squared so  - X - = +

'PRINT #25, " ydif "; ydif;
IF ydif = latcoord THEN GOTO enddtg   'airportcoord(apci) must = 0 exit loop

apci = apci + 1

'PRINT #25, " ap lon coord "; airportcoord(apci)

xdif = AirPortCorr * (loncoord - airportcoord(apci))

'PRINT #25, " xdif "; xdif
DisToGo = (xdif ^ 2 + ydif ^ 2) ^ .5

dtgi1 = dtgi1 + 1
distogoa(dtgi1) = DisToGo

GOTO startdtg

enddtg:
apci = 0
dtgi1 = 0

EndDisToGo:
ON ERROR GOTO 0
'PRINT #25, " Finished calculating distances to go "; TIMER
'PRINT #25, "FRE(-2) stack size available"; FRE(-2)
```

```
'TIMER ON
RETURN

edistancetogo:
'PRINT #25, " dtgi1 "; dtgi1
'PRINT #25, " apci "; apci
'PRINT #25, " xdif "; xdif
'PRINT #25, " ydif "; ydif

CLS
PRINT " Distance to go error. "
CLOSE
END

'EP***************    ERASE PLANE
erasePlane:
 headingpr = -1 * heading
 phorzpos = winsize - lonmin * scale
 pvertpos = winsize - LatMin * scale
 PSET (phorzpos, pvertpos)         'erase horz, vert
'IF ByPassFlag$ = "BP" THEN GOTO skiperaseplane
 DRAW "C0"
 GOSUB plotpixplane
'skiperaseplane:
RETURN

'PP********************* PLOT PLANE ***************************
plotplane:

headingpr = -1 * heading
phorzpos = winsize - lonmin * scale
pvertpos = winsize - LatMin * scale

IF ByPassFlag$ = "BP" THEN GOTO skipplotplane

PSET (phorzpos, pvertpos)         'horz, vert
DRAW "C15"
GOSUB plotpixplane

skipplotplane:
RETURN

plotpixplane:
DRAW "TA=" + VARPTR$(headingpr) + "L10 D1 R20 U1 L10 U5 D15 L4 R8 L4 U15 D5"
RETURN



'DI******************* DATA INVALLID
datainvalid:
'PRINT #25, "data invalid"; TIME$
TIMER STOP
LOCATE 1, 60
BEEP
'               1
'      12345678901234567890
PRINT "DATA INVALID";
delaytime1 = TIMER
```

```
di2:
delaytime2 = TIMER

IF INKEY$ = CHR$(27) THEN GOTO endit              'esc to exit
IF delaytime2 < delaytime1 + 2 THEN GOTO di2

              '               1
              '      12345678901234567890
LOCATE 1, 60: PRINT "            "

TIMER ON
'END
RETURN



'NDCI******************* NO DATA COMING IN
nodatacomingin:
ndcicnt = ndcicnt + 1
IF ndcicnt > 24 THEN GOTO endit

IF AirportDatScrFlag$ = "aptdatscr" THEN GOTO nodatfltinfo

'PRINT #25, "no data coming in "; TIMER
BEEP
TIMER STOP
'                     123456789012345678
LOCATE 1, 60: PRINT "NO DATA COMING IN";
delaytime1 = TIMER
delayndci:
IF INKEY$ = CHR$(27) THEN GOTO endit              'esc to exit
delaytime2 = TIMER
IF delaytime2 < delaytime1 + 4 THEN GOTO delayndci
'                     123456789012345678
LOCATE 1, 60: PRINT "                 ";
TIMER ON
GOTO endgarminsub

nodatfltinfo:
'PRINT #25, "no data coming in flt info"; TIMER
BEEP
TIMER STOP
'                     123456789012345678
LOCATE 57, 30: PRINT "NO DATA COMING IN";
delaytime1 = TIMER

delayndcifi:
IF INKEY$ = CHR$(27) THEN GOTO endit              'esc to exit
delaytime2 = TIMER
IF delaytime2 < delaytime1 + 4 THEN GOTO delayndcifi
'                     123456789012345678
LOCATE 57, 30: PRINT "                 ";
TIMER ON
GOTO endallcalc

nomatchfound:
'CLS
'CLS 1
'LOCATE 15, 1: PRINT "                No match found "
```

```
GOTO endprt51

errorbigsearch:
'PRINT #25, "  error big search "
RESUME endprt51

opentemp19err:
CLS
PRINT "open temp19 error"
CLOSE
END



'*****************    PLOT LINE
plotline:
 xc1 = winsize - ((VAL(MID$(field2$, 4, 6))) * scale)
  yc1 = winsize - ((VAL(MID$(field3$, (cp + 1), 6))) * scale)
 xc2 = winsize - ((VAL(MID$(field4$, 4, 6))) * scale)
 yc2 = winsize - ((VAL(MID$(field5$, (cp + 1), 6))) * scale)
LINE (yc1, xc1)-(yc2, xc2)  'again this works ass backwards fro the way you
RETURN

'*****************       PLOT CIRCLE
plotcircle:        'Don't dick with this! It is very tricky indeed.
 xc1 = winsize - ((VAL(MID$(field2$, 4, 6))) * scale)
 yc1 = winsize - ((VAL(MID$(field3$, (cp + 1), 6))) * scale)
  radius = (VAL(field4$)) * scale
 angle1 = (VAL(field5$))
 IF angle1 <= 90 THEN angle2 = 360 + (angle1 - 90) ELSE angle2 = angle1 - 90
 angle2 = 360 - angle2
 startc = (angle2 / 57.295)
 angle3 = (VAL(field6$))
 IF angle3 <= 90 THEN angle4 = 360 + (angle3 - 90) ELSE angle4 = angle3 - 90
 angle4 = 360 - angle4
 endc = (angle4 / 57.295)         '* (-1)
'                 ,       , ,start, end      Normally
CIRCLE (yc1, xc1), radius, , endc, startc  'Tricky, start and end are reversed
RETURN

'******************* END IT AS A RESULT OF ESC KEY PRESS
endit:
TIMER OFF
CLS 0
CLOSE
'SHELL "erase temp19.dat"    'kill don't work don't use it
'SHELL "erase temp45.dat"
END

'****************** END of PROGRAM
```