```
'  LATEST REVISED ... December 30th 2018

'  PRE-PROGRAM NOTATIONS

'  Nothing fancy or elaborate here. Just some simple fundamentals to learn from and build on.

'  This is a demonstration of --

'  1. Raising Numbers to the Power of 3 - IE, 3x3x3 is 27 - An Example of "Cubed" Numbers.

'  2. Placing commands for various tasks into specific SUBs and FUNCTIONs.

'  3. Creating User Menus and Prompts, along with supporting on-screen displays.

'  4. Custom creation of FUNCTIONs to assist in program flow, control and appearance.

'     In this demonstration program, the two key FUNCTIONs are as follows ....

'     a. MasterAll -- Control of major program sequences (Main Menu; File Saving;
'                     print out to hard copy or PDF; exit from program)

'     b. NNQuiry -- Serves as a common menu choice selector for different parts of
'                   this demo program, utilizing pre-fixed number codes to generate
'                   specific actions and responses.

'  5. Custom creation of SUBs that control on-screen displays and output to
'     other devices. Two key SUBs in this demonstration are as follows ...

'     a. DaMainDisplay - Generates on-screen menus and supporting information

'     b. ShowDaResults - Generates the KUBE data for on-screen display, as well
'                        as for file saving and/or print out.

'  6. Custom creation of SUBs for file saving (KnameDaFile, DoFileSave and
'     DaSaveFileMenu) and output to printer or PDF (DoPrintOut)

'  7. A further note about FUNCTION NNQuiry

'     In addition to using single letters (F P S G X etc) for specific actions, usage is also
'     made of certain non-alphabet and non-numeric board keys for demonstration purposes. These
'     particular keys are represented in the form of special CHR$ codes. For example, the Escape
'     (ESC) key is CHR$(27), while the F-1 Key is CHR$(0,59). Following is a listing of available
'     CHR$ codes for the non-alpha/non-number keys.

'     CHR$ Code    KEY             CHR$ Code    KEY             CHR$ Code    KEY

'     8            Backspace       0,63         F-5             0,77         Arrow Right
'     9            TAB             0,64         F-6             0,72         Arrow Up
'     13           Return/Enter    0,65         F-7             0,80         Arrow Down
'     27           Escape (ESC)    0,66         F-8             0,71         Home
'     32           Space Bar       0,67         F-9             0,79         End
'     0,59         F-1             0,68         F-10            0,73         Page Up
'     0,60         F-2             0,87         F-11            0,81         Page Down
'     0,61         F-3             0,88         F-12            0,82         Insert
'     0,62         F-4             0,75         Arrow Left      0,83         Delete

'     NOTE 1 -- You could use "Cursor" in place of "Arrow"

'     NOTE 2 -- Any F-Key that has been programmed to return specific characters rather than its
'               default CHR$ code can not be used as responses to INKEY$ or WAITKEY$ inquiries.
'               For example, if the F-8 key is programmed as follows ....

'               KEY 8, "EXIT"+CHR$(13)

'               .... then that particular F-key will not return its default CHR$ code. To restore
'               the particular F-key to its default state, do as follows ...

'               KEY 8, ""

#COMPILE EXE "..\EXEs\Kube-0-Seven.exe"

#COMPILER PBCC 5
' The code herein is compatible with PowerBASIC Console Compiler Version 5.00 or higher

' If compiling with an earlier version of PBCC, use that version number instead. Also,
' be sure to look within this demo program for notations about lines needing to be
' changed for compatibility with earlier PBCC versions.


#DIM ALL

' #INCLUDE "win32api.inc" ''' For special calls and functions. Was not needed in this demo.

' If not compiling with PBCC Version 5.00 or later, then as per the demands of your
' particular compiler you may be required to DECLARE each of the FUNCTIONs and SUBs in
' this program. All such DECLAREs are placed at this point in the coding. A DECLARE line
' starts with DECLARE and then all of the exact syntax of the particular SUB or FUNCTION's
' first line.

' Two examples .....
```

```
'       DECLARE FUNCTION AT0(BYVAL RRow AS LONG, BYVAL RCol AS LONG) AS STRING

'       DECLARE SUB DaMainDisplay (GeeMode AS LONG, GeeXtra AS LONG)

' The DECLARE'd SUB and/or FUNCTION will reside elsewhere in the program

' NOTE -- FUNCTION PBMAIN usually does not require a DECLARE.

 FUNCTION PBMAIN () AS LONG

    ' Console and cursor modes set here
    CONSOLE SET LOC 160,20
    CONSOLE SET SCREEN 50,80
    CONSOLE SET VIEW 50, 80
    CURSOR OFF

' In addition to these settings, it is recommended that the user adjusts MS-WINDOWS Shortcut
' Properties for the console window displaying Kube-O-Seven as follows ...

' Compatibility -- Windows XP (Service Pack 3)
' Font -- Lucida Console .. Size: 18 .. Check Bold On
' Screen Buffer -- Width: 88 .. Height: 48 .. Uncheck Wrap Text feature
' Window Size -- Width: 88 .. Height: 48
' Location -- Left: 10 .. Top: 10 .. Uncheck Let System Position feature

    CALL MasterAll(1,"Begin"):' Action trigger that will get program started

    CALL FirstMenu
' Official Starting Point of Main Program. Control returns here
' when user presses an "X" in any of the menu subs herein.
' Console, Color and Cursor Modes re-set before leaving program
    CONSOLE SET SCREEN 25,80
    COLOR 7,0
    CURSOR ON

END FUNCTION


 FUNCTION MasterAll (GeeMode AS LONG, WWhatText AS STRING) AS STRING
    STATIC MMasterText AS STRING

    '   MasterAll is a Master Control Function in this Program

    '   Value of
    '   MMasterText             Action
    '   -------------------------------------------
    '   Begin                   Start of Program
    '   Main Refresh            Restore Main Menu Display
    '   File-1 (File-2, File-3) File Save
    '   Print-1                 Print Out
    '   Gag                     Microsoft Gag
    '   END                     Exit out of the Program
    '   EXIT NOW                Forces Unconditional exit from program

    IF GeeMode=2 THEN
       MasterAll=MMasterText
       EXIT FUNCTION
    END IF

    IF GeeMode=1 THEN
       MMasterText=WWhatText
    END IF

 END FUNCTION


 FUNCTION AT0(BYVAL RRow AS LONG, BYVAL RCol AS LONG) AS STRING
    ' on-screen cursor placement
    LOCATE RRow, RCol
 END FUNCTION


 FUNCTION AT2(BYVAL RRow AS LONG, BYVAL RCol AS LONG, BYVAL RRGB AS LONG) AS STRING
    ' on-screen cursor placement with foreground color
    IF RRow<>99 AND RCOL<>99 THEN
       ' Allows use of this function with AT2(99,99,xx) calls when only a
       ' color type is neeeded.
       LOCATE RRow, RCol
    END IF

    COLOR RRGB
 END FUNCTION


 FUNCTION AT4(BYVAL FRGnd AS LONG, BYVAL BKGnd AS LONG) AS STRING
    ' on-screen foreground and background color
    COLOR FRGnd, BKGND
 END FUNCTION
```

```
FUNCTION FolderExist(FileToTest AS STRING) AS LONG
   ' checks for existence of a folder on a specified drive.
   ' See also SUBs FolderTest, KnameDaFile and DoFileSave
   LOCAL Dummy&
   Dummy& = GETATTR(FileToTest)
   FUNCTION = (ERRCLEAR = 0)
END FUNCTION


SUB FolderTest (Foldz AS LONG,ChekName AS STRING)
    LOCAL Test AS STRING, CantDo AS LONG
    ' See also FUNCTION FolderExist, as well as SUBs KnameDaFile and DoFileSave

    SELECT CASE Foldz
      CASE 1
          Test="C:\Temp"
          ' Check for existence of folder (Directory) TEMP on Drive C. Only need the one \ here

      CASE 2
          Test=ChekName:' A user-created folder (directory) on the drive of their choice
          IF LEN(Test)<6 THEN EXIT SUB:' foldername to be at least 3 characters

          ' Case 2 not actually used in this demo, but is included to illustrate where-to-save options

      CASE ELSE: EXIT SUB
    END SELECT

    IF FolderExist(Test)=-1 THEN
    ' -1 is YES, it exists - no further action needed
    ELSEIF FolderExist(Test)=0 THEN
      MKDIR Test
    ' 0 is no, it does not exist, so create it
    END IF

END SUB


FUNCTION NNQuiry (GeeMode AS LONG) AS LONG
   LOCAL XzitKode AS LONG
   LOCAL NQzitor AS STRING
   STATIC YYourInput AS LONG

   IF GeeMode=2 THEN
      NNQuiry=YYourInput
      EXIT FUNCTION
   END IF


   IF GeeMode=1 THEN
      XzitKode=11

      LLOOPER:
      NQzitor=WAITKEY$
      NQzitor=UCASE$(NQzitor)

      IF MasterAll(2,"")="Begin" OR MasterAll(2,"")="Main Refresh" THEN
          IF NQzitor="F" THEN XzitKode=20
          IF NQzitor="P" THEN XzitKode=21
          IF NQzitor=CHR$(9) THEN XzitKode=22
          ' CHR$(9) is the Tab Key.
          ' This hidden choice is included for DEMO purposes
      END IF

      IF MasterAll(2,"")="Print-1" THEN
          IF NQzitor="Y" THEN XzitKode=30
      END IF

      IF MasterAll(2,"")="File-2" OR MasterAll(2,"")="File-3" THEN
          IF NQzitor="S" THEN XzitKode=20
      END IF

      IF MasterAll(2,"")="Gag" THEN
          IF NQzitor=CHR$(32) OR NQzitor=CHR$(8) THEN XzitKode=88
          IF NQzitor=CHR$(0,77) THEN XzitKode=99
          ' CHR$(32) is the Space Bar

          ' CHR$(8) is the Backspace Key, and CHR$(0,77) is the Arrow (Cursor) Right Key.
          ' These hidden choices are included for DEMO purposes.
      END IF

      IF XzitKode=11 AND MasterAll(2,"")<>"Gag" THEN
          IF NQzitor="G" OR NQzitor=CHR$(0,68) THEN XzitKode=88
          IF NQzitor="X" OR NQzitor=CHR$(0,88) THEN XzitKode=99
          ' CHR$(0,68) is F-10 key .... CHR$(0,88) is F-12 Key
          ' F-10 and F-12 are hidden choices included for DEMO purposes

          ' Using MasterAll(2,"")<>"Gag" condition restricts the Microsoft Gag
          ' exit options to the Space Bar, Backspace or Arrow Right Keys
      END IF

      IF XzitKode=11 THEN GOTO LLOOPER
```

```
            YYourInput=XzitKode
            ' Stores XzitKode value in STATIC variable YYourInput
            ' for future retrieval
    END IF

END FUNCTION



SUB FirstMenu
    LOCAL GGoDoThis, KKounter AS LONG

    IF MasterAll(2,"")="EXIT NOW" THEN EXIT SUB

    GGoDoThis=5

    IF MasterAll(2,"")="Begin" THEN
        CLS
        COLOR 0,0
        FOR KKounter=1 TO 49
            ? AT0(KKounter,1);REPEAT$(78," ")
        NEXT KKounter

        CALL MasterAll(1,"Main Refresh")
        GGoDoThis=1
        CALL ShowDaResults(1)

    END IF

    IF MasterAll(2,"")="Main Refresh" THEN
        GGoDoThis=1
        CLS
        CALL ShowDaResults(2)
        CALL DaMainDisplay(1,0)

    END IF

    IF GGoDoThis=1 THEN
        CALL NNQuiry(1)

        SELECT CASE NNQuiry(2)
            CASE 20
                CALL MasterAll(1,"File-1")
                CALL KnameDaFile
                GGoDoThis=2

            CASE 21
                CALL MasterAll(1,"Print-1")
                CALL DoPrintOut
                GGoDoThis=2

            CASE 22
                CALL MasterAll(1,"Gag")
                CALL MicroSoftAlarm
                GGoDoThis=2

            CASE 88
                CALL MasterAll(1,"Main Refresh")
                CALL FirstMenu
                GGoDoThis=2

            CASE 99
                CALL MasterAll(1,"END")
                GGoDoThis=5

        END SELECT

    END IF


    IF GGoDoThis=5 OR MasterAll(2,"")="END" THEN

        CLS
        ? AT2(4,2,11);"Thank you for using ";

        COLOR 15,0: ? "Kube-0-Seven"

        SLEEP 1500:' Pause of 1.50 seconds

        CALL MasterAll(1,"EXIT NOW")
        EXIT SUB
        ' Unconditional Program Exit Forced
    END IF

END SUB


SUB KnameDaFile
    LOCAL LegalCharacters, IllegalCharacters, YourInput AS STRING
```

```
LOCAL GGoDoThis AS LONG
' See also FUNCTION FolderExist, as well as SUBs FolderTest and DoFileSave

GGoDoThis=5

SELECT CASE MasterAll(2,"")
   CASE "File-1"
        GGoDoThis=1
        YourInput=""
        CLS
        CALL ShowDaResults(2)
        CALL DaMainDisplay(2,1)

   CASE ELSE: EXIT SUB

END SELECT


LLooper:

   IF GGoDoThis=1 THEN
      ? AT2(34,50,15);"";
      LINE INPUT YourInput
   END IF

   YourInput=UCASE$(YourInput)
   ' Lower-case letters made upper-case, because character-stripping procedure below
   ' would take out any lower-cases.

   LegalCharacters="0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ-"

   IllegalCharacters=REMOVE$(YourInput, ANY LegalCharacters):' isolate any non-legal character
   IF LEN(IllegalCharacters)>=1 THEN YourInput=REMOVE$(YourInput, ANY IllegalCharacters)

   ' If at least one illegal character exists, it will be removed here

   SELECT CASE YourInput
      CASE "FILE","SAVE","222","2222","22222","222222"
           CALL MasterAll(1,"File-3")
           CALL DaSaveFileMenu
           GGoDoThis=2
           ' Send program to the 2nd File Save Menu

      CASE "RETRY","333","3333","33333","333333"
           CALL MasterAll(1,"File-1")
           GGoDoThis=1
           YourInput=""

           CLS
           CALL ShowDaResults(2)
           CALL DaMainDisplay(2,1)
           ' Try again. Not really needed, but include for demo purposes


      CASE "CANCEL","USER","111","1111","11111","111111","888","8888","88888","888888"
           CALL MasterAll(1,"Main Refresh")
           GGoDoThis=8
           ' Send program back to Main Menu User Options.

      CASE "END","EXIT","999","9999","99999","999999"
           CALL MasterAll(1,"END")
           GGoDoThis=9
           ' Unconditional exit from the program.

   END SELECT

   IF GGoDoThis=1 THEN
      IF LEN(YourInput) >=5 AND LEN(YourInput) <=25 THEN GGoDoThis=7
      ' Successful filename entry. Exit Loop and perform actions in IF GGoDoThis=7 block below
   END IF


   IF GGoDoThis=1 THEN
      YourInput=""
      CALL DaMainDisplay(2,2)
      ' No successful file entry or alternate choice, so back to input line
   END IF

IF GGoDoThis=1 THEN GOTO LLooper


IF GGoDoThis=8 AND MasterAll(2,"")="Main Refresh" THEN CALL FirstMenu:' Return to Main Menu

IF GGoDoThis=7 THEN
   CALL MasterAll(1,"File-2")
   CALL DoFileSave(1,YourInput)

   YourInput=""
   CALL DaSaveFileMenu
END IF
```

```
      END SUB


SUB DoFileSave (GeeMode AS LONG, YourInput AS STRING)
      LOCAL KKounter, XzitKode AS LONG
      STATIC FFullName, DaDriveDir, FFileKname AS STRING
      ' See also FUNCTION FolderExist, as well as SUBs FolderTest and KnameDaFile

      SELECT CASE MasterAll(2,"")
        CASE "File-2": XzitKode=1

        CASE ELSE: XzitKode=2

      END SELECT

      IF XzitKode=2 THEN EXIT SUB

      IF GeeMode=1 THEN
          DaDriveDir="C:\TEMP":' Only need the one \ here

          CALL FolderTest(1,DaDriveDir)
          '   Checks for Existence of a user-specified folder (directory).
          '   If the folder does not already exist, SUB FolderTest creates it.
          '   C:\TEMP's existence check test that is already set up.
          '   If checking for folders on C or other drives, the numerical value would be 2 or more,
          '   and the string parameter would be the folder and drive - say, F:\Kube - in question.

          FFileKname=YourInput
          ' saving filename in STATIC variable FFileKname for current and future retrieval

          FFullName=BUILD$(DaDriveDir,"\",FFileKname,".TXT")


'   SPECIAL NOTE -- For those using earlier versions of PowerBASIC Console Compiler

'       The FFullName=BUILD$ line shown above is valid only in PBCC Versions 5.00
'       and above. For earlier PBCC versions, change the line as follows ...

'       FFullName=DaDriveDir + "\" + FFileKname + ".TXT"


          OPEN FFullName FOR OUTPUT AS #1
          CALL ShowDaResults(3):' Save results stored in SUB ShowDaResults to a named file
          CLOSE #1

      END IF

      ' If SUB DoFileSave is called again, the value of GeeMode can be any number other than 1,
      ' and the value of YourInput can be a null string.  As a result, only the lines shown
      ' below will be executed.

      COLOR 0,15

      FOR KKounter=30 TO 35
          ? AT0(KKounter,47);STRING$(31,32):' Establish White Background Display
      NEXT KKounter

      ? AT0(31,48);"Results saved to file"
      ? AT0(33,48);FFileKname+".TXT"
      ? AT0(34,48);"in folder ";DaDriveDir
      COLOR 7,0

END SUB


SUB DaSaveFileMenu
      LOCAL GGoDoThis AS LONG

      GGoDoThis=2

      SELECT CASE MasterAll(2,"")
        CASE "File-2", "File-3"
            GGoDoThis=1
            CLS
            CALL ShowDaResults(2)
            CALL DaMainDisplay(2,3)
            IF MasterAll(2,"")="File-2" THEN CALL DoFileSave(2,"")

        CASE ELSE: EXIT SUB

      END SELECT

      LLooper:
        IF GGoDoThis=1 THEN CALL NNQuiry(1)

        SELECT CASE NNQuiry(2)
          CASE 20
              CALL MasterAll(1,"File-1")
              CALL KnameDaFile
              GGoDoThis=2
```

```
                    ' S/20 - Go to File Save Instructions and File Name Entry


            CASE 88,99
                    IF NNQuiry(2)=88 THEN CALL MasterAll(1,"Main Refresh")
                    IF NNQuiry(2)=99 THEN CALL MasterAll(1,"END")
                    CALL FirstMenu
                    GGoDoThis=2
                    ' G/88 - Go back to Main User Options Menu
                    ' X/99 - Exit Program

         END SELECT

     IF GGoDoThis=1 THEN GOTO LLooper

END SUB


SUB DoPrintOut
     LOCAL GGoDoThis AS LONG

     GGoDoThis=2

     SELECT CASE MasterAll(2,"")
       CASE "Print-1"
            GGoDoThis=1
            CLS
            CALL ShowDaResults(2)
            CALL DaMainDisplay(3,1)

       CASE ELSE: EXIT SUB

     END SELECT


     LLooper:

      IF GGoDoThis=1 THEN CALL NNQuiry(1)

        SELECT CASE NNQuiry(2)
          CASE 30
                    ' Y/30 - printing to printer, PDF file or other

                    CALL DaMainDisplay(3,2):' Printing In Progress
                    ERRCLEAR
                    XPRINT ATTACH CHOOSE:' Attach and Choose a Printer thru WINDOWS

                    XPRINT SET ORIENTATION 1
                    ' Sets to Portrait (8.5 x 11 Letterhead)

                    XPRINT FONT "Courier New", 14, 1
                    ' Choose the Font, Font Size (14 pts) and Style (1 = Bold)

                    IF ERR=0 AND LEN(XPRINT$)>0 THEN
                       CALL ShowDaResults(4):'     Will print out info stored in SUB ShowDaResults
                       XPRINT FORMFEED:'            Issue a formfeed
                       XPRINT CLOSE:'               Detach the printer
                    END IF

                    CALL DaMainDisplay(3,3)
                    SLEEP 1550
                    ' Printing Completed. Program control will stay within this particular loop
                    ' until options G/88 or X/99 (see below) are selected.

          CASE 88,99
                    IF NNQuiry(2)=88 THEN CALL MasterAll(1,"Main Refresh")
                    IF NNQuiry(2)=99 THEN CALL MasterAll(1,"END")
                    CALL FirstMenu
                    GGoDoThis=2
                    ' G/88 - Go back to Main User Options Menu
                    ' X/99 - Exit Program

         END SELECT

     IF GGoDoThis=1 THEN GOTO LLooper

END SUB


SUB DaMainDisplay (GeeMode AS LONG, GeeXtra AS LONG)
     LOCAL KKounter, JColor, NXAA, NXBB AS LONG
     LOCAL WFAA AS STRING
     DIM JXTRA(3) AS LONG

     JXTRA(1)=99
     ' In GeeMode CASEs 2 and 3 (see below) the value of JXTRA(1) will be changed from 99 to 1.
     ' This will make active the IF JXTRA(1)=1 block that follows the entire SELECT CASE GeeMode
     ' block in this particular SUB (see below).

     JXTRA(2)=99
     ' In GeeMode CASE 3 (see below) the value of JXTRA(2) will be changed from 99 to 1 for one
```

```
' text display, and then from 99 to 2 for another text display.


SELECT CASE GeeMode
    CASE 1,2,3
        ' common to all three GeeMode cases, treated here rather than below
        WFAA=CHOOSE$(GeeMode,"Main ","File Save ","Print Out ")
        ? AT2(19,47,15);"User Options - ";WFAA
END SELECT


SELECT CASE GeeMode
    CASE 1
        ' User Options Menu
        ? AT2(21,47,11);"Press A Letter        "
        ? AT2(23,47,15);"F -";AT2(23,51,11);"Save Results to File"
        ? AT2(25,47,15);"P -";AT2(25,51,14);"Print Out Results   "
        ? AT2(27,47,15);"X -";AT2(27,51,11);"Exit This Program   "


    CASE 2
        ' File Save Instructions
        SELECT CASE GeeXtra
            CASE 1
                ? AT2(21,47,14);"Follow all the instructions"
                ? AT0(22,47);"as shown below."

                ? AT0(42,03);"Legal Characters to use are A to Z  a to z  0 to 9  and the hyphen -   "
                ? AT0(43,03);"Type your desired characters in the space after >> above, then press   "
                ? AT0(44,03);"the ENTER Key. FILENAME.TXT will be saved to the folder TEMP on Drive-C"
                ? AT0(45,03);"If the TEMP folder does not already exist, Kube-0-Seven will create it."
                ? AT2(47,03,15);"Enter FILE or SAVE to recall the Save File Menu -OR- enter CANCEL or   "
                ? AT0(48,03);"USER to Return to User Options. Enter END or EXIT to end this program. "

            CASE 2
                COLOR 7,0
                FOR KKounter=30 TO 35
                    ? AT0(KKounter,46);STRING$(32,32)
                NEXT KKounter
                ' clear out any previously entered text


            CASE 3
                ' File Save Menu
                COLOR 7,0
                ? AT2(21,47,11);"Press A Letter           "
                ? AT2(23,47,15);"S -";AT2(23,51,11);"Save to File        "

                JXTRA(1)=1:' Makes active the IF JXTRA(1)=1 block (see below)

        END SELECT

        IF GeeXtra<3 THEN
            ' Common Text for GeeMode CASE 2 only when GeeXtra value is less then 3
            ? AT2(30,47,11);"Enter Filename"
            ? AT2(32,47,11);"(5 to 25 characters)"
            ? AT2(34,47,15);">>"
        END IF


    CASE 3
        ' Printer Menu and Instructions
        SELECT CASE GeeXtra
            CASE 1
                ? AT2(21,47,11);"Press A Letter            "
                ? AT2(23,47,15);"Y -";AT2(23,51,11);"Print It"

                JXTRA(1)=1:' Makes active the IF JXTRA(1)=1 block (see below)

                ? AT2(30,47,14);"Pressing Y invokes standard "
                ? AT0(31,47);"WINDOWS Printer Menu. Select"
                ? AT0(32,47);"desired printer and options,"
                ? AT0(33,47);"then print your document.   "

            CASE 2
                COLOR 0,15
                FOR KKounter=30 TO 38
                    ? AT0(KKounter,47);STRING$(30,32)
                    ' Establish White Background Display
                NEXT KKounter

                ? AT0(31,49);"Printing In Progress      "
                ? AT0(33,49);"Press Y for another copy,"
                ? AT0(34,49);"or make another choice.  "

            CASE 3
                COLOR 0,15
                ? AT0(31,49);"Printing Completed    "
                COLOR 7,0:' reset colors

        END SELECT
```

```
                IF GeeXtra<3 THEN JXTRA(2)=GeeXtra
                ' Makes active the IF JXTRA(2)=1 or 2 block (see below)
                ' only when GeeXtra value is less than 3.

        END SELECT


        IF JXTRA(1)=1 THEN
            ' Called via JXTRA(1) variable change from 99 to 1 in GeeMode
            ' CASEs 2 and 3 above

            ? AT2(25,47,15);"G -";AT2(25,51,14);"User Options - Main "
            ? AT2(27,47,15);"X -";AT2(27,51,11);"Exit This Program   "
        END IF

        IF JXTRA(2)=1 OR JXTRA(2)=2 THEN
            ' Called via JXTRA(2) variable change from 99 to 1 -OR-
            ' from 99 to 2 in GeeMode CASE 3 above

            COLOR 14,0: NXAA=47
            IF JXTRA(2)=2 THEN COLOR 0,15: NXAA=49
            ? AT0(36,NXAA);"Prints to 8.5 x 11 Letter"
            ? AT0(37,NXAA);"Courier New, 14 pt, BOLD "
            COLOR 7,0:' reset colors
        END IF


END SUB



SUB ShowDaResults (GeeMode AS LONG)
    LOCAL KKount, JColor, FCount, NXAA, NXBB AS LONG
    LOCAL WFAA, WFBB AS STRING
    DIM Header(8) AS STATIC STRING, BBODY(31) AS STATIC STRING
    DIM KKubee(101) AS LONG
    DIM KKNumber(101) AS STATIC STRING, KKubex(101) AS STATIC STRING
    DIM SideBar (10) AS STATIC STRING

    ' Variable GeeMode controls the action within this SUB

    ' In GeeMode CASE 1 below, Header and Body info will be compiled, then stored in STATIC strings.
    ' This allows for such data to be displayed on-screen (GeeMode CASE 2), saved to a text file
    ' (GeeMode CASE 3) or printed out to paper-PDF-OTHER (GeeMode CASE 4) when appropriate calls
    ' are made back to this SUB.


    SELECT CASE GeeMode
        CASE 1
            ' Compiling Header Info - See GeeMode CASE 2 (Display) for handling of Header(2)
            ' This information will be stored in the STATIC Header(x) strings for later
            ' retrieval in GeeMode CASEs 2, 3 and 4

            '        123456789a123456789b123456789c123456789d123456789e1234567  MID$,1,14  MID$,15,42 - Header(2)
            Header(2)="  Kube-O-Seven :: Powerbasic Console Compiler v5.05 Demo"
            Header(4)="  Numbers Cubed (Raised to the power of 3, see examples)"
            Header(6)="  No.  Cubed     No.   Cubed     No.   Cubed     No.   Cubed"
            Header(7)="  "+STRING$(55,45):' The hyphen - strip below "No. Cubed" line


            ' Compiling Main Body Info
            ' This information will be stored in the STATIC BBODY(x) strings for later
            ' retrieval in GeeMode CASEs 2, 3 and 4

            FOR KKount=1 TO 99
                KKNumber(KKount)=USING$("##",KKount)+"  "

                KKubee(KKount)=KKount^3

                SELECT CASE KKubee(KKount)
                    CASE      1 TO      9: WFAA="      #"
                    CASE     10 TO     99: WFAA="     ##"
                    CASE    100 TO    999: WFAA="    ###"
                    CASE   1000 TO   9999: WFAA=" #,###"

                    CASE 10000 TO 99999
                        WFAA="##,###"
                        IF KKount >= 31 THEN WFAA=" ##,###"

                    CASE 100000 TO 999999: WFAA="###,###"
                END SELECT

                KKubex(KKount)=FORMAT$(KKubee(KKount),WFAA)

            NEXT KKount

            SideBar(1)="EXAMPLES"
            SideBar(2)="64 is 4 cubed"
            SideBar(3)="4 x 4 x 4"
            SideBar(4)="729 is 9 cubed"
```

```
            SideBar(5)="9 x 9 x 9"
            ' Sidebar info will be dealt with in GeeMode CASEs 2-3-4

            WFBB="     "

            FOR KKount=1 TO 30
                BBody(KKount)=BUILD$("   ",KKNumber(KKount),KKubex(KKount),WFBB,KKNumber(KKount+30), _
                              KKubex(KKount+30),WFBB,KKNumber(KKount+60),KKubex(KKount+60))

                IF KKount <=9 THEN
                    BBody(KKount)=BUILD$(BBody(KKount),WFBB,KKNumber(KKount+90),KKubex(KKount+90))
                END IF

            NEXT KKount


'  SPECIAL NOTE -- For those using earlier versions of PowerBASIC Console Compiler
'
'          The two BBody(KKount)= lines shown just above this notation would need to be changed
'          because the Build$ function is available only to those using PBCC Version 5.00 and above.
'
'          The lines to put in are as follows ....
'
'          BBody(KKount)="  " + KKNumber(KKount) + KKubex(KKount) + WFBB + KKNumber(KKount+30) _
'                        + KKubex(KKount+30) + WFBB + KKNumber(KKount+60) + KKubex(KKount+60)
'
'          BBody(KKount)=BBody(KKount) + WFBB + KKNumber(KKount+90) + KKubex(KKount+90)


        CASE 2
            '   On-Screen Display of Info

            ' Displaying Header Info
            FOR KKount=1 TO 7
                JColor=15
                IF KKount=4 THEN JColor=14

                SELECT CASE KKount
                    CASE 2
                        ? AT2(KKount,1,JColor);MID$(Header(2),1,14)
                        ? AT2(KKount,15,11);MID$(Header(2),15,42)
                        ' MID$ function allows on-screen display of Header(2)
                        ' info at different points w/different colors

                    CASE 4,6,7
                        ? AT2(KKount,1,JColor);Header(KKount)
                END SELECT
            NEXT KKount


            ' Displaying Main Body Info
            FOR KKount=1 TO 30
                SELECT CASE KKount
                    CASE  1: FCount=0:' for control of line breaks in display of KUBE information
                    CASE 11: FCount=1
                    CASE 21: FCount=2
                END SELECT

                ' From Row 8 (7+1+0) to 17 (7+10+0), then 19 (7+11+1) to 28 (7+20+1),
                ' then 30 (7+21+2) to 39 (7+30+2)

                JColor=15
                SELECT CASE KKount
                    CASE 5,10,15,20,25,30: JColor=14
                END SELECT

                ? AT2(7+KKount+FCount,03,JColor);KKNumber(KKount);KKubex(KKount)
                ? AT2(7+KKount+FCount,17,JColor);KKNumber(KKount+30);KKubex(KKount+30)
                ? AT2(7+KKount+FCount,32,JColor);KKNumber(KKount+60);KKubex(KKount+60)

                IF KKount <= 9 THEN
                    IF KKount=9 THEN JColor=14
                    ? AT2(7+KKount+FCount,47,JColor);KKNumber(KKount+90);KKubex(KKount+90)
                END IF

            NEXT KKount

            ' On-screen, the EXAMPLES SideBar info is displayed next to the 91 thru 99 KUBE
            ' results because space is needed elsewhere for displaying menu and instruction text.
            NXAA=4
            FOR NXBB=1 TO 5
                NXAA=NXAA+2:' On-screen lines (rows) 6, 8, 10, 12 and 14
                ? AT2(NXAA,61,15);SideBar(NXBB)
            NEXT NXBB


        CASE 3, 4
            ' GeeMode CASE 3 - Save to File  (PRINT #1)
            ' File is Named in SUB KnameDaFile
            ' File is Opened and Closed in SUB DoFileSave
```

```
                    ' GeeMode CASE 4 - Print Out Results (XPRINT)
                    ' Choice of printer, print font options and XPRINT Attach,
                    ' Formfeed and Close all done in SUB DoPrintOut

                    ' Header Info saved to file (GeeMode 3) or printed out (GeeMode 4)
                    FOR KKount=1 TO 7
                        IF GeeMode=3 THEN PRINT #1, Header(KKount)
                        IF GeeMode=4 THEN XPRINT Header(KKount)
                    NEXT KKount

                    ' Main Body and Sidebar Info saved to file (GeeMode 3) or printed out (GeeMode 4)

                    NXBB=0
                    ' NXBB value must be set to zero outside of KKount FOR-NEXT Block. This allows for
                    ' proper value increase (from 0 to 1, then 1 to 2, etc. up to 5) in the SELECT
                    ' CASE KKount (CASEs 11, 13, 15, 17, 19) scenario located below.

                    FOR KKount=1 TO 30
                        IF GeeMode=3 THEN PRINT #1, BBody(KKount);
                        IF GeeMode=4 THEN XPRINT BBody(KKount);

                        ' BBody(KKount) immediately followed by a semi-colon ; to allow for the EXAMPLES SideBar info
                        ' to be placed next to the 71 thru 80 KUBE results (KKount 11, 13, 15, 17, 19). The line
                        ' breaks occurring at KKount 10 (KUBE 10-11/40-41/70-71) and KKount 20 (KUBE 20-21/50-51/80-81)
                        ' are accounted for, as well as the ending of the remaining lines of KUBE info.

                        SELECT CASE KKount
                          CASE 11,13,15,17,19
                                NXBB=NXBB+1
                                ' NXBB value increase must occur here. The result is ...
                                ' KKount 11 - NXBB=1 / KKount 13 - NXBB=2 / KKount 15 - NXBB=3
                                ' KKount 17 - NXBB=4 / KKount 19 - NXBB=5
                                ' Allows SideBar(1) to (5) strings to be properly handled.

                                IF GeeMode=3 THEN PRINT #1, "     ";SideBar(NXBB)
                                IF GeeMode=4 THEN XPRINT "      ";SideBar(NXBB)

                          CASE 10,20
                                IF GeeMode=3 THEN PRINT #1, "": PRINT #1, ""
                                IF GeeMode=4 THEN XPRINT "": XPRINT ""
                                ' Line breaks (10-11, 40-41, 70-71 | 20-21, 50-51, 80-81)

                          CASE ELSE
                                IF GeeMode=3 THEN PRINT #1, ""
                                IF GeeMode=4 THEN XPRINT ""
                                ' All other lines of KUBE Info

                        END SELECT

                    NEXT KKount

            END SELECT

END SUB



SUB MicroSoftAlarm
        LOCAL GeeWhizBill, WFAA AS STRING
        LOCAL GGoDoThis, KKount, JColor, MKount AS LONG

        SELECT CASE MasterAll(2,"")
          CASE "Gag"
                CLS
                CALL ShowDaResults(2)

          CASE ELSE: EXIT SUB

        END SELECT

        FOR KKount=19 TO 27
            ? AT0(KKount,30);STRING$(48,32);' Blank out portion of KUBE display for Gag.
        NEXT KKount

        SLEEP 0400:' Pause of 0.4 seconds
        ? AT2(20,32,15);"WHAT !?"

        SLEEP 0210:' Pause of 0.21 seconds
        MKount=0
        FOR KKount=41 TO 57 STEP 2

            MKount += 1


'   SPECIAL NOTE -- For those using earlier versions of PowerBASIC Console Compiler

'        The MKount += 1 line, an example of compound operations allowed in PBCC Versions
'        5.00 and above, would need to be changed for earlier versions, as follows ...

'        MKount=MKount + 1
```

```
        IF MKount>9 THEN EXIT FOR

        GeeWhizBill="MICROSOFT"
        JColor=CHOOSE(MKount,11,14,11,10,07,10,11,14,11):' Colors chosen for each letter in MICROSOFT
        WFAA=MID$(GeeWhizBill,MKount,1):' Each letter extracted and displayed

        SLEEP 0210:' Pause of 0.21 seconds
        ? AT2(20,KKount,JColor);WFAA

    NEXT KKount

    SLEEP 0210:' Pause of 0.21 seconds
    ? AT2(20,59,7);"..?";

    SLEEP 0700:' Pause of 0.7 seconds
    COLOR 15: ? "  PLEASE !!"

    SLEEP 1055:' Pause of 1.055 seconds

    ? AT2(23,32,11);"What kind of a ";
    COLOR 14: ? CHR$(173);"!";CHR$(155);"$%^& ";
    COLOR 11: ? "fool do you think I am?"

    SLEEP 1055:' Pause of 1.055 seconds

    ? AT2(26,32,11);"Press the ";
    COLOR 14: ? "SPACE BAR ";
    COLOR 11: ? "to make another choice"

    GGoDoThis=1

    LLooper:

        IF GGoDoThis=1 THEN CALL NNQuiry(1)

        SELECT CASE NNQuiry(2)
            CASE 88,99
                IF NNQuiry(2)=88 THEN CALL MasterAll(1,"Main Refresh")
                IF NNQuiry(2)=99 THEN CALL MasterAll(1,"END")
                CALL FirstMenu
                GGoDoThis=2
                ' CHR$(32) Space Bar/88 - Go back to Main User Options Menu
                ' X/99 - Exit Program
        END SELECT

    IF GGoDoThis=1 THEN GOTO LLooper

END SUB
```