Alpha Count

This module is part of the original MMBasic library. It is reproduced here with kind permission of Hugh Buckle and Geoff Graham. Be aware it may reference functionality which has changed or is deprecated in the latest versions of MMBasic.

```
       ' ALPHA COUNT PROGRAMME
       ' This counts the frequency of each alphabetic character in a text
document
       ' and the frequency of double letters.
       ' Counts are output to a CSV file which can be read by Excel and Open
Office Calc.
       ' If the input count file is overwritten as output, counts can be
accumulated from several text documents.
       ' On completion it draws graphs of the percentage of each letter and
double letter.
     Dim LetterCounts(26)
     Dim OLDCounts(26)
     Dim NewCounts(26)
     Dim DblLetterCounts(26)
     Dim OldDblCounts(26)
     Dim NewDblCounts(26)
     Dim a(26)
     Dim pLine$(4)

'Mainline
     GoSub Initialise        ' Initialise
     Timer=0
     GoSub LetterCounts      ' Accumulate letter counts from input text file
     For i = 1 To 26         ' check that text file contains some letters
       atot = atot + LetterCounts(i)
     Next
     If atot = 0 Then
       Print
       Print TxtFileName$ " contains no alphabetic characters."
     Else
       GoSub AddNewToOldCounts ' Add new counts to old
       GoSub PrintGraphs       ' Print accumulated counts graphically
       GoSub WriteCounts       ' Write out csv file with accumulated counts
     EndIf
     Print Timer
     GoTo WindUp             ' Close files and exit

Initialise:
       ' Intialise values in case the routine is rerun
     Cls
     For i = 1 To 26
       LetterCounts(i) = 0
       DblLetterCounts(i) = 0
```

```
     Next
     LastChar$ = ""
     Print @(42,0) "ALPHANUM.BAS Counts occurrences of each letter and
double letter"
     Print @(24,12) "in an input text file, graphs the counts and writes
them to a CSV file."
     Print @(42,24) "Text file counts will be added to those from an input
.CSV file."
     Print

GetInputTextFileName:
       ' Ask for input text file name - Reply 'Exit' stops the program
       ' If filename doesn't have an extension, .TXT is assumed

     Input "Input text filename (.txt assumed) - 'Exit  ' to exit: ",
TxtFileName$
     If LCase$(TxtFileName$) = "exit" Then GoTo WindUp
       ' IF file name doesn't have an extension, add .txt
     If Instr(1, TxtFileName$, ".") = 0 Then TxtFileName$ = TxtFileName$ +
".txt"
     Option error continue
     Open TxtFileName$ For INPUT As #1
     If MM.Errno <> 0 Then
       Print TxtFileName$ " doesn't exist"
       Print
       GoTo GetInputTextFileName
     EndIf
     Option error abort

GetOldCountFileName:
       ' Get the old count file names - Reply 'None' if there isn't one
       ' If filename doesn't have an extension, .CSV is assumed

     Input "Old count file name (.csv assumed) - 'none  ' if none: ",
OldCountFileName$
     If LCase$(OldCountFileName$) = "none" Then GoTo GetNewCountFileName
     If LCase$(OldCountFileName$) = "exit" Then GoTo WindUp
       ' IF file name doesn't have an extension, add .CSV
     If Instr(1, OldCountFileName$, ".") = 0 Then OldCountFileName$ =
OldCountFileName$ + ".csv"
     Option error continue
     Open OldCountFileName$ For input As #2
     If MM.Errno <> 0 Then
       Print OldCountFileName$ " doesn't exist. Enter 'none', 'exit  ' or
another file"
       Print
       GoTo GetOldCountFileName
     EndIf
     Option error abort
       ' Load counts from Old Count File
     GoSub LoadOldCounts
```

```
GetNewCountFileName:
       ' Create a new output count file - Reply 'Exit' stops the program
       ' If a filename extension is not provided, .CSV is appended
       ' If this filename is same as input count filename, check that it is
OK to overwrite

     Input "New count file name (.csv assumed) - 'Exit  ' to exit: ",
NewCountFileName$
     If LCase$(NewCountFileName$) = "exit" Then GoTo WindUp
     If Instr(1, NewCountFileName$, ".") = 0 Then NewCountFileName$ =
NewCountFileName$ + ".csv"
       ' If old file exists, ask if it should be replaced. If not, get
another filename
     Option error continue
     Open NewCountFileName$ For input As #3
     If MM.Errno = 0 Then
       Print "OK to overwrite "+OldCountFileName$+" Y/N";:Input ""; Reply$
       If LCase$(Left$(Reply$,1)) <> "y" Then
         Print
         GoTo GetNewCountFileName
       EndIf
       Close #3
     EndIf
     Option error abort
       ' Don't open it until the old counts have been read in case the old
count file is to be overwritten
     Return

LetterCounts:
       ' Read characters one by one from the text input file and convert to
lower case
       ' Set start position of the progress line (which will show that
something is happening)

     x = 20: y = 120
     Do While Not Eof(1)
         ' Save previous character to compare with next to accumulate double
character count
       LastChar$ = NextChar$
         ' Get next character
       NextChar$ = Input$(1, #1)
         ' If upper case character, convert to lower case
       NextChar$ = LCase$(NextChar$)
         ' If a letter, accumulate totals for that letter, else ignore it
       If NextChar$ >= "a" And NextChar$ <= "z" Then
         GoSub AccumulateCounts
       EndIf
         ' periodically show progress
       If CharCount > 500 Then
         GoSub PrintProgressLine
         CharCount = 0
```

```
      Else
        charCount = CharCount + 1
      EndIf
    Loop
    Close #1
    Return

PrintProgressLine:
      ' Print a progress line to show that something is happening every
500th character read
      ' prints another dot and also toggles the MM Power LED

    If ProgressDots < 70 Then
      ProgressDots = ProgressDots + 1
      ProgressLine$ = ProgressLine$ + "."
      Print @(x,y) ProgressLine$
        ' Toggle the power LED on and off
      Pin(0) = ProgressDots Mod 2
    Else
      Print @(x,y) Space$(70)
      ProgressDots = 0
      ProgressLine$ = ""
    EndIf
    Return

AccumulateCounts:
      ' Accumulate counts of each letter and double letters.

    i = Asc(NextChar$) - 96   ' Note: ASC("a") = 97
    LetterCounts(i) = LetterCounts(i) + 1
      ' If same as last letter, accumulate double letter count
    If NextChar$ = LastChar$ Then DblLetterCounts(i) = DblLetterCounts(i) +
1
    Return

AddNewToOldCounts:
      ' Add new counts to old

    For i = 1 To 26
      NewCounts(i) = OldCounts(i) + LetterCounts(i)
      NewDblCounts(i) = OldDblCounts(i) + DblLetterCounts(i)
    Next
    Return

PrintGraphs:
      ' Print the 2 graphs - Proportion of single letters and double
letters
      ' Do this twice;
      '   1st for current text file and then
      '   if there was an input .CSV file, for the accumulated counts
```

```
      'Common graph parameters
     bars = 26    ' number of bars in the graph
     xscale = 1   'Use full screen width of 480 pixels
     yScale = 0.5 'Use half screen height of 432/2 = 216 pixels
       ' set ends of x axis, allowing space to the left of the graph for
scale
     xorig = 5*6         ' Indent y axis 5 characters
     xmax  = 480*xScale  ' end of x axis
       ' set ends of y axis, allowing space above and below graph
       ' for title & x scale
     yTopSpace = 2*12    ' 2 lines
     yBotSpace = 2*12    ' 2 lines

       ' Get y values for single letters in the input text file
     atot = 0
     For i = 1 To bars
       a(i) = LetterCounts(i)
       atot = atot +a(i)
     Next
       ' Print the graph in top half of screen
     Cls
     Title$ = "Percent of single letters in " + TxtFileName$
     GoSub SetupSingleLetterGraphParms
     GoSub PrintAGraph

       ' Get y values for double letters in the input text file
     atot = 0
     For i = 1 To bars
       a(i) = DblLetterCounts(i)
       atot = atot +a(i)
     Next

     If atot > 0 Then
         ' Print the graph in bottom half of screen (yoffset sets bottom
half)
       Title$ = "Percent of double letters in " + TxtFileName$
       GoSub SetupDoubleLetterGraphParms
       GoSub PrintAGraph
     Else
       Print
       Print "There are no double letters in the text file."
       Print
     EndIf

       ' If the input .CSV file was overwritten, then repeat the graphs for
accumulated totals
       ' otherwise we are finished
     If NewCountFileName$ = OldCountFileName$ Then
       Print "Press A to see accumulated percentages; Any other to exit";
       Do
         Reply$ = Inkey$
```

```
      Loop Until Reply$ <> ""
      If LCase$(Reply$) <> "a" Then
        Return
      EndIf
    Else
      End
    EndIf

      ' Get y values for single letters in the accumulated counts file
    atot = 0
    For i = 1 To bars
      a(i) = NewCounts(i)
      atot = atot +a(i)
    Next
      ' Print the graph in top half of screen
    Cls
    Title$ = "Percent of single letters in " + NewCountFileName$
    GoSub SetupSingleLetterGraphParms
    GoSub PrintAGraph

      ' Get y values for double letters in the accumulated counts file
    atot = 0
    For i = 1 To bars
      a(i) = NewDblCounts(i)
      atot = atot +a(i)
    Next

    If atot > 0 Then
        ' Print the graph in bottom half of screen (yoffset sets bottom
half)
      Title$ = "Percent of double letters in " + NewCountFileName$
      GoSub SetupDoubleLetterGraphParms
      GoSub PrintAGraph
    Else
      Print
      Print "There are no double letter in the acumulated counts file."
      Print
    EndIf

    Print "Press A to see the first graph again; any other to exit";
    Do
      Reply$ = Inkey$
    Loop Until Reply$ <> ""
    If LCase$(Reply$) = "a" Then
      GoTo PrintGraphs
    EndIf

    Return

SetupSingleLetterGraphParms:
    yorig = 432*yScale - yTopSpace - yBotSpace
```

```
     ymax  = yTopSpace   ' top of y axis

     Return

SetupDoubleLetterGraphParms:
     yOffset = 432/2
     yorig = 432*yscale - yTopSpace - yBotSpace + yOffset
     ymax  = yTopSpace + yOffset  ' top of y axis

     Return

PrintAGraph:
      ' Prints a graph at preset points
      ' Normalise all the values to percent of total letters read
     HiY = 0
     For i = 1 To bars
       a(i) = a(i) / atot * 100
       If a(i) > HiY Then HiY = a(i)
     Next

       ' Set top of y axis to highest value of y + 10% and round up
     HiY = HiY*1.1
     HiY = Cint(HiY/5)*5

       ' Set origin for the graph and the scale
       ' Normal font chars occupy 6x12 pixels giving 36 lines of 80 chars
each
       ' Calc No of pixel per data unit
     xUnit = (xmax-xorig)/bars
     yUnit = (ymax-yorig)/HiY

       ' Draw Axes
     Line (Xorig,ymax)-(Xorig,yorig)
     Line (Xorig,yorig)-(xmax,yorig)

       ' Draw tick marks on y axis and print tick values
     tickmks = 5   ' Int(HiY/10+.5)
     ytickspace = (yOrig-ymax)/tickmks
     For i = 0 To tickmks-1
       Line (xOrig,ymax+i*ytickspace) - (xOrig+5,ymax+i*ytickspace)
       Print @(0,ymax+i*ytickspace-6) Hiy-i*Hiy/tickmks
     Next

       ' Print title, centred
       ' Title chars are 6 px wide
     Print @((xmax-xOrig)/2-Len(Title$)*6/2,ymax-12) Title$

       ' Print alphabet below x axis and percentages below that
     Print @(12,yOrig+24) "%"
     barspace = Fix((xmax-xOrigin)/(bars+1))
     For i = 0 To bars-1
```

```
       Print @(xOrig+barspace*i+8, yOrig+12) Chr$(i+1+64)
       Print @(xOrig+barSpace*i+2, yOrig+24) Int(a(i+1)+0.5)
     Next

       ' Print graph bars
     For i = 1 To bars
       x1 = xOrig+barspace*(i-1)+6
       y1 = yOrig
       x2 = x1+barspace/2
       y2 = yOrig-a(i)*(yorig-ymax)/Hiy
       Line (x1,y1)-(x2,y2),1,BF
     Next

     Return

LoadOldCounts:
        ' Subroutine: Load counts from old counts file
        ' Being a .CSV file, concatenate numeric characters
        ' until a comma is found then
        ' load the resulting number into the array.

      Do While Not Eof(2)
          ' First line contains single letter counts
        Line Input #2, Counts$
        StartPos = 1
        LetterNo = 1
        Do
          CommaPos = Instr(StartPos,Counts$,",")
          OldCounts(LetterNo) = Val(Mid$(Counts$,StartPos,CommaPos-
StartPos))
          LetterNo = LetterNo + 1
          StartPos = CommaPos + 1
        Loop Until StartPos >= Len(Counts$)

          ' Second line contains double letter counts
        Line Input #2, Counts$
        StartPos = 1
        LetterNo = 1
        Do
          CommaPos = Instr(StartPos,Counts$,",")
          OldDblCounts(LetterNo) = Val(Mid$(Counts$,StartPos,CommaPos-
StartPos))
          LetterNo = LetterNo + 1
          StartPos = CommaPos + 1
        Loop Until StartPos >= Len(Counts$)
      Loop
      Close #2
      Return

WriteCounts:
        ' Subroutine: Write new counts to output as .CSV file
```

```
        ' If the file exists, delete it
        ' Writes 2 lines of comma-separated numbers
        '   1st line contains the counts of each alphabetic character
        '   2nd line, counts of each double alphabetic character

      Open NewCountFileName$ For OUTPUT As #3
      For i = 1 To 26
        Print #3, Format$(NewCounts(i),"%0g") ",";
      Next
      Print #3
      For i = 1 To 26
        Print #3, Format$(NewDblCounts(i),"%0g") ",";
      Next
      Close #3
      Return

WindUp:
       ' Exit the program after restoring normal error handling
      Option error continue
      Print Timer
      End
```