Module Module1

'------------------------------------------------------------

'- File Name : Module1.vb -

'- Part of Project: WordAnalysis\_Dang -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: January 29, 2022 -

'------------------------------------------------------------

'- File Purpose: -

'- -

'- This file contains the main program that will process

'- all the text from the input file then later print it to

'- output file

'------------------------------------------------------------

'- Program Purpose: -

'- -

'- The purpose of this program is to analyze the text input file

'- which enter by user, read it, then analyze it, later it will read the output

'- file path entered by user. User will have option to print out to the console

'-

'------------------------------------------------------------

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'--- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS --- SUBPROGRAMS ---

'-----------------------------------------------------------------------------------

Sub Main()

'------------------------------------------------------------

'- Subprogram Name: Main -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: January 29, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine make sure that user is can interact with the console\

'- get input from the user then later check if user want to see the anylysis '-

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically):

'- strPath -input file path from use

'- strReport - Report file path from use

'- strAnswer - yes/no input from user

'------------------------------------------------------------

'Set up the Console Screen

Console.WindowWidth = 100 'width console is 100

Console.Title = "Word Analysis Profiler Application"

Console.ForegroundColor = ConsoleColor.Blue 'set letter to vlue

Console.BackgroundColor = ConsoleColor.White 'set background to white

Console.WriteLine("Please enter the path and name of the file to process:")

'Request the path to txt file

Dim strPath As String = Console.ReadLine() 'request input from user

If pathValidation(strPath) = False Then

GoTo ExitConsole

End If

Console.WriteLine("Processing Completed...")

Console.WriteLine()

Console.WriteLine("Please enter the path and name of the report file to generate:")

Dim strReport = Console.ReadLine() 'take input to string report

If pathDuplication(strPath, strReport) Then 'call function to check if the file path is

GoTo ExitConsole 'duplicate to the input file. else, it will take

End If ' another input

If pathValidation(strReport) = False Then ' call function path validation to

GoTo ExitConsole 'check in while loop, if not valid, it will take one more

End If

Analyze(strPath, strReport) 'call analyze subprogram for reading and print out

Console.WriteLine()

'write report

Console.Write("Report File Generation Completed...")

Console.WriteLine()

Console.WriteLine()

Console.WriteLine()

Console.Write("Would you like to see the report file? [y/n]")

Console.WriteLine()

Dim strAnswer = Console.ReadLine().ToUpper 'taking input if user want to print

'out the analysis or not.All input up to uppder case

'checking the input value if it is valid or not

If strAnswer = "Y" Then

printStats(strReport) 'if yes, calling printstats function

ElseIf strAnswer = "N" Then

Console.WriteLine() 'if no don't print anything from the file

Console.WriteLine("Thank you for using Word Analysis!")

Console.WriteLine("Application has completed. Press any key to end.")

Console.Read() 'wait for user to enter to quit the program

Else

ExitConsole:

Console.WriteLine("Wrong input!")

Console.WriteLine("Please restart the application.")

Console.Read() 'wait for user to enter to quit the program

End If

End Sub

Function pathDuplication(ByVal strPath As String, ByVal strReport As String) As Boolean

'------------------------------------------------------------

'- Subprogram Name: pathDuplication -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: January 29, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine make sure that user is not writing the duplicate output file with the input file

'-by passing in 2 two string that user's input

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): None

If strReport.Equals(strPath) = True Then

Dim blnDuplicate = True

Return blnDuplicate

End If

End Function

Function pathValidation(ByVal strPath As String) As Boolean

'------------------------------------------------------------

'- Subprogram Name: pathValidation -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: January 29, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine make sure that user is input the right value

'- else it will check and allow user to enter another input

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): None

If String.IsNullOrWhiteSpace(strPath) Then 'if the input is white or nothing

Console.WriteLine()

Return False

ElseIf Not strPath.EndsWith(".txt") Then 'check if the file is txt or ot

Console.WriteLine()

Return False

ElseIf Dir(strPath) = "" Then ' Check if file exist

Console.WriteLine()

Return False

Else 'else continue to the loops

Console.WriteLine()

Return True

End If

Return False

End Function

Sub Analyze(strPath As String, strReport As String)

'------------------------------------------------------------

'- Subprogram Name: Analyze -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: January 30, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- Allow user to Analyze the input text file and then

'-print it out to the output file with desire path by pass in by Val

'Using objReader and Writer

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically):

'arrFileContents - Array that contains all the list of word

'arrHighestKey - array of word that appear at the most amount

'arrLowestKey - arrray of word that appear the least ampont

'arrWord - a refined list of arrFileContents make sure there are no duplicate

'dblUtilization - calculate how much word is utilized

'dctWord - a dictionary that keep a key (string) with a value (times appear)

'intHighestValue - the highest time that that word has appeared

'intLongestWord - the longest word count for padding when printing out report

'intLowestValue - the least time word appear

'intMaxStarLine - check how many stars are on the line so there are no enter down a line

'intStarDraw - times

'intTotalValue -how many word is in the text file

'intValue - How many time that word appear

'objReader - obj for reading file

'objWriter - obj for writing on file

'strHighestKey - The word that appear the highest time

'strLowestKey - The word that appear the least

'strPrintHighest - A string that print out all the highest appear words

'strPrintLowest - A string that print out all the lowest appear words

'strWord - a word in the collection

'initiate object Reader

Dim objReader As System.IO.StreamReader

objReader = System.IO.File.OpenText(strPath) 'take in the pass in value of the file path

Dim arrFileContents() As String 'create an array then let the reader read through and split

arrFileContents = objReader.ReadToEnd.Split(" "c)

'dictionary that keep the value and an array of word for easy sorting

Dim dctWord As IDictionary(Of String, Integer) = New Dictionary(Of String, Integer)()

Dim arrWord As ArrayList = New ArrayList()

'Add word to dictionary

'Loop through the array file contents to all element

For i = 0 To arrFileContents.Count() - 1

'Create a string word that make all the word to uppercase

'and remove the comma and period

Dim strWord = arrFileContents(i).Replace(",", "")

strWord.Trim()

strWord = strWord.Replace(".", "")

strWord = strWord.ToUpper()

strWord = Replace(Replace(strWord, Chr(10), ""), Chr(13), "")

'if the string is not empty then add the word to dictionary and array

'if the word repeat, find the word in the dictionary copy the key and value

'then add the new one in with value +1

If Not strWord = "" Then

If dctWord.ContainsKey(strWord) Then

Dim intValue = 0

dctWord.TryGetValue(strWord, intValue)

intValue = intValue + 1

dctWord.Remove(strWord)

dctWord.Add(strWord, intValue)

Else

arrWord.Add(strWord)

dctWord.Add(strWord, 1)

End If

End If

Next

'close the file

objReader.Close()

'sort array alphabetically

arrWord.Sort()

'initaiate obj writer

Dim objWriter As System.IO.StreamWriter

objWriter = My.Computer.FileSystem.OpenTextFileWriter(strReport, False)

Dim intTotalValue As Integer ' total times word appear

'Declare the highest key word count and value

Dim strHighestKey = arrWord(0).ToString

Dim intHighestValue = 0

dctWord.TryGetValue(strHighestKey, intHighestValue)

Dim strLowestKey = arrWord(0).ToString

Dim intLowestValue = 0

dctWord.TryGetValue(strHighestKey, intLowestValue)

Dim intLongestWord = 0

'Find the Highest/Lowest Word with highest/Lowest Value

'Also find the longest word for easier padding

For i = 1 To arrWord.Count - 1

Dim strKey = arrWord(i).ToString

Dim intValue = 0

Dim intWordLength = arrWord(i).ToString.Length

dctWord.TryGetValue(strKey, intValue)

If (intValue > intHighestValue) Then

intHighestValue = intValue

strHighestKey = strKey

ElseIf (intValue < intLowestValue) Then

intLowestValue = intValue

strLowestKey = strKey

End If

If intWordLength > intLongestWord Then

intLongestWord = intWordLength

End If

Next

'Create 2 array keeping the word that have the same amount of appearance

Dim arrHighestKey As ArrayList = New ArrayList()

Dim arrLowestKey As ArrayList = New ArrayList()

'loop go through eveey word in dictionary

'then if it the same with the highest/lowest, it wil be added

For i = 1 To arrWord.Count - 1

Dim strKey = arrWord(i).ToString

Dim intValue = 0

dctWord.TryGetValue(strKey, intValue)

If (intValue = intHighestValue) Then

arrHighestKey.Add(strKey)

ElseIf (intValue = intLowestValue) Then

arrLowestKey.Add(strKey)

End If

Next

'Start writing report to report file

objWriter.WriteLine(vbTab + vbTab + vbTab + "Word Analysis Statistics")

objWriter.WriteLine()

objWriter.WriteLine("There were a total of " + arrWord.Count().ToString + " unique words encountered.")

objWriter.WriteLine()

'Loop go through every word in the sorted array

'Then print out the word, follow by the times it appear and a histogram

For i = 0 To arrWord.Count - 1

Dim intValue = 0

Dim intMaxStarLine As Integer

dctWord.TryGetValue(arrWord(i).ToString, intValue)

intTotalValue = intTotalValue + intValue

If intLongestWord > 15 Then

objWriter.Write(arrWord(i).ToString.PadRight(intLongestWord) + String.Format(" : {0:x4} ", intValue).PadRight(0))

intMaxStarLine = Console.BufferWidth - intLongestWord - 10 '(padding)Longest Word + 10(space)

Else

objWriter.Write(arrWord(i).ToString.PadRight(15) + String.Format(" : {0:x4} ", intValue).PadRight(0))

intMaxStarLine = Console.BufferWidth - 25 '(padding)15 + 10(space)

End If

'check to see how many star should draw

Dim intStarDraw = intValue \* intMaxStarLine / intHighestValue

For j = 0 To intStarDraw

objWriter.Write("\*")

Next

objWriter.WriteLine()

Next

objWriter.WriteLine()

'calculate word utilization

Dim dblUtilization As Double = CDbl(intTotalValue) / CDbl(arrWord.Count)

objWriter.WriteLine("Average Word Utilization: " + dblUtilization.ToString)

'A string that using a function that return a string of array

Dim strPrintHighest = printArr(arrHighestKey)

Dim strPrintLowest = printArr(arrLowestKey)

objWriter.WriteLine("Highest Word(s) Utilization: " + intHighestValue.ToString + " on " + strPrintHighest)

objWriter.WriteLine("Lowest Word(s) Utilization: " + intLowestValue.ToString + " on " + strPrintLowest)

objWriter.Close() 'close and save the file

End Sub

Function printArr(arr As ArrayList) As String

'------------------------------------------------------------

'- Subprogram Name: printArr -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: January30, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine read the array then which later append it to the string

'-which later return that string

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- strPrint - Print string -

'------------------------------------------------------------

Dim strPrint As String = ""

'a for loop that run through the array which later

'append to the strPrint with no dangling comma

For i = 0 To arr.Count - 1

If i = arr.Count - 1 Then

strPrint = strPrint + arr(i).ToString

Else

strPrint = strPrint + arr(i).ToString + ", "

End If

Next

Return strPrint

End Function

Sub printStats(strReport As String)

'------------------------------------------------------------

'- Subprogram Name: printStats -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: January 29, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine read the path name to the report file then print it out to the console screen

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): -

'- objReader - object For reading file -

'------------------------------------------------------------

Dim objReader As System.IO.StreamReader

objReader = System.IO.File.OpenText(strReport) ' create obj reader to the file

Console.WriteLine()

'If still not end of file. print out the console

While Not (objReader.EndOfStream)

Console.WriteLine(objReader.ReadLine())

End While

objReader.Close()

Console.WriteLine()

Console.WriteLine("Application has completed. Press any key to end.")

Console.Read()

End Sub

End Module