|  |  |  |
| --- | --- | --- |
|  |  | Information and communication technology  SBA Report  Topic: Composition Analyzer  **Name: HO YIN HEI**  **Class: 6E**  **Class No: 11** |

Contents

[Introduction 3](#_Toc147269420)

[Algorithm design 3](#_Toc147269421)

[Finding the frequency letter 3](#_Toc147269422)

[Finding the frequency of word 3](#_Toc147269423)

[Calculation of the total number of words 3](#_Toc147269424)

[Calculation of the total number of sentences 3](#_Toc147269425)

[Calculation of the total number of paragraphs 3](#_Toc147269426)

[Calculation of the total number of function word 3](#_Toc147269427)

[The library used in the program 3](#_Toc147269428)

[Conclusion 4](#_Toc147269429)

[Source code 4](#_Toc147269430)

[Acknowledgement 4](#_Toc147269431)

[Reference 4](#_Toc147269432)

# Introduction

(At least 50 words)

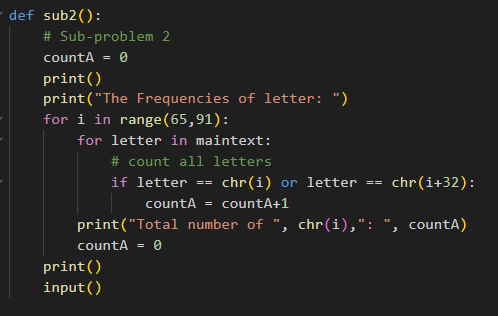
* My program provides several functions, such as counting the total number of words, characters, letters, sentences, paragraphs, function words and finding the frequency of a specific word.
* A screenshot of a computer program

  Description automatically generated
* The above image is the screen capture of the menu.
* Briefly describe the following paragraphs

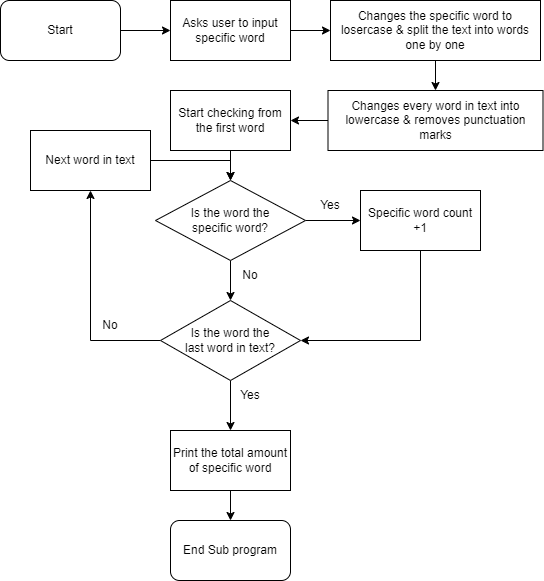
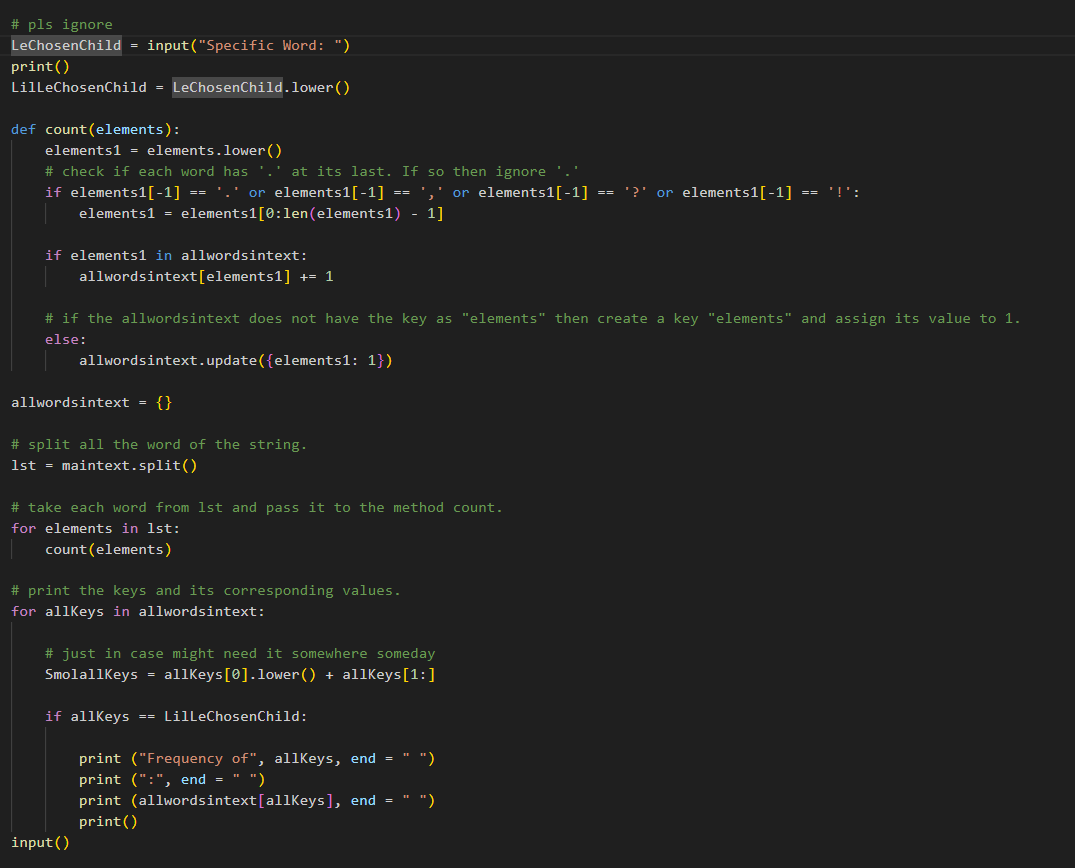
# Algorithm design

## Finding the frequency letter

* A screenshot of a computer program

  Description automatically generated
* This program will calculate the frequency of letters inside your text from the start of the text.
* Explanation on how it works: The counting value will start with 0. It selects from chr(65) (which is the letter “a”) to chr(91) (which is the letter “z”) and check every letter in your text. If the letter matches, the counting value will add 1 and then move on to the next letter on the text. When all letters inside the text are checked, it will print the total number of the letter which it is currently counting and then move on to the next letter and check if any letters in the text matches until the letter “z” has also finished counting.
* 

## Finding the frequency of word

* 
* This program will calculate the total number of a specific word inside your text from the start of the text.
* Explanation on how it works: First, it asks the user to input the specific word that they want to find. Next, it changes the specific word into lowercase and split the text into words one by one. Then, it changes every word in the text into lowercase and removes the punctuation marks. After that, it counts the amount of words that matches the specific word. Finally, it prints out the results.
* 

## Calculation of the total number of words

* A screenshot of a computer

  Description automatically generated
* This program will calculate the total number of words inside your text from the start of the text.
* Explanation on how it works: First, it splits up the text into a lot of single words. Next, it counts the number of single words that’s been split from the text. Finally, it prints out the amount of single words that’s been split from the text.
* A computer screen shot of text

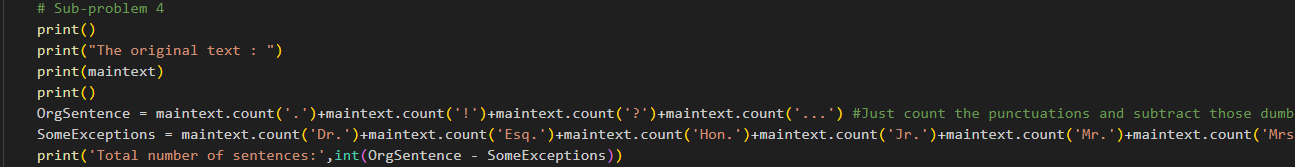
  Description automatically generated

## Calculation of the total number of sentences

* A screenshot of a computer screen

  Description automatically generated

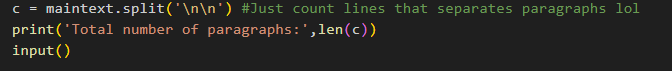
End Sub program

* This program will calculate the total number of words inside your text from the start of the text.
* Explanation on how it works: First, it counts the number of full stops, question marks or any symbols which ends the sentence. Next, it counts the number of special cases that also uses the symbols but does not end the sentence such as “Dr.” and “Mr.” After that, the program adds up the symbols and then subtract the number of special cases that does not count as a sentence ending symbol. Finally, it prints out the number of sentences in the text.
* 

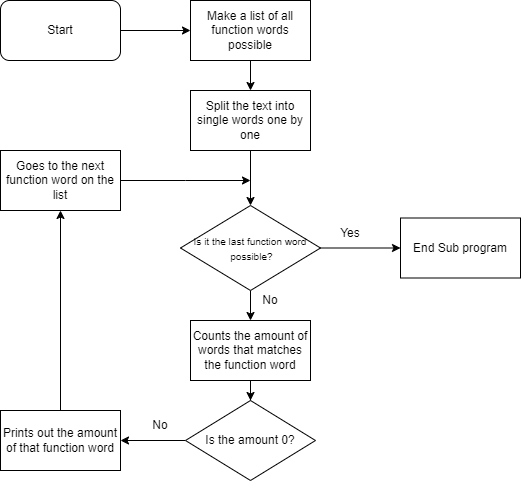
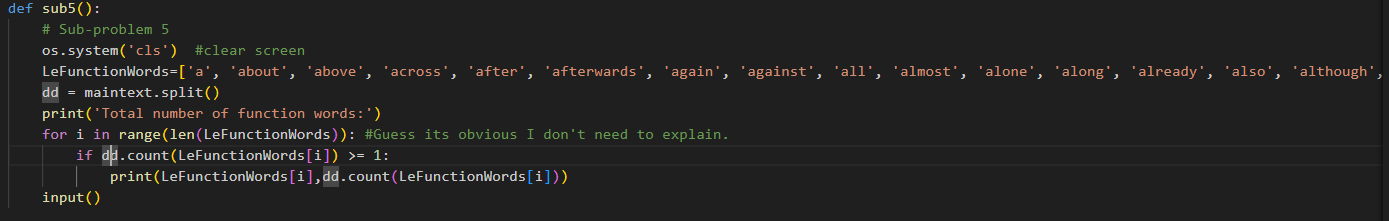
## Calculation of the total number of paragraphs

A black and white squares with black text

Description automatically generated

* This program will calculate the total number of paragraphs in the text.
* Explanation on how it works: First, it splits the text by separating lines since we usually start a new column for a new paragraph. Then, it counts the number of texts split.
* 

## Calculation of the total number of function word

* Function word means a word whose purpose is to contribute to the syntax rather than the meaning of a sentence, for example, in “We do not live here”, “do” is the function word.
* 
* This program will calculate the total number of function words in the text.
* Explanation on how it works: First, it made a table of most possible function words that would occur in the text. Then, it loops through all the function words in the list and then count the amount of that specific function word appears in the text. If the number of that function word appears once or more than once, the program prints out the amount of that function word appears on the text.
* 

## The library used in the program

* os library
* The os library is a built-in os module with methods for interacting with the operating system, like creating files and directories, management of files and directories, input, output, environment variables, process management, etc. It is a simpler way to execute things than other libraries.

# Conclusion

* While making this program, I’ve learnt to use libraries and some common knowledge of programming. I’ve also improved my debugging skills after encountering several setbacks such as changing the way how users typing in text into reading a .txt file inside the user’s desktop. This is quite a big challenge since I’ve never done something similar before. I hope I could have more chances to improve my programming skills and do some more complicated tasks.

# Source code

|  |
| --- |
| import os #import os for library  import time    #maintext = input("Write something...") # No need anymore cuz we pro and use file instead of typing    def sub1():  # Sub-problem 1  print()  print("The original text : ")  print(maintext)  print()    # using split() to count words in string  res = len(maintext.split())    print("Number of words : " + str(res))  print("Number of characters : ", len(maintext))  print()  input()    def sub2():  # Sub-problem 2  countA = 0  print()  print("The Frequencies of letter: ")  for i in range(65,91):  for letter in maintext:  # count all letters  if letter == chr(i) or letter == chr(i+32):  countA = countA+1  print("Total number of ", chr(i),": ", countA)  countA = 0  print()  input()    def sub3():  # Sub-problem 3  # input specific word UI  os.system('cls') #clear screen  print()  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*YLMASS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Composition Analyzer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*Please input a specific word\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')    # pls ignore  LeChosenChild = input("Specific Word: ")  print()  LilLeChosenChild = LeChosenChild.lower()    def count(elements):  elements1 = elements.lower()  # check if each word has '.' at its last. If so then ignore '.'  if elements1[-1] == '.':  elements1 = elements1[0:len(elements1) - 1]    if elements1 in allwordsintext:  allwordsintext[elements1] += 1    # if the allwordsintext does not have the key as "elements" then create a key "elements" and assign its value to 1.  else:  allwordsintext.update({elements1: 1})    allwordsintext = {}    # split all the word of the string.  lst = maintext.split()    # take each word from lst and pass it to the method count.  for elements in lst:  count(elements)    # print the keys and its corresponding values.  for allKeys in allwordsintext:    # just in case might need it somewhere someday  SmolallKeys = allKeys[0].lower() + allKeys[1:]    if allKeys == LilLeChosenChild:    print ("Frequency of", allKeys, end = " ")  print (":", end = " ")  print (allwordsintext[allKeys], end = " ")  print()  input()    def sub4():  # Sub-problem 4  print()  print("The original text : ")  print(maintext)  print()  OrgSentence = maintext.count('.')+maintext.count('!')+maintext.count('?')+maintext.count('...') #Just count the punctuations and subtract those dumb special cases that made those punctuation marks doesnt count as the end of a sentence.  SomeExceptions = maintext.count('Dr.')+maintext.count('Esq.')+maintext.count('Hon.')+maintext.count('Jr.')+maintext.count('Mr.')+maintext.count('Mrs.')+maintext.count('Ms.')+maintext.count('Messrs.')+maintext.count('Mmes.')+maintext.count('Msgr.')+maintext.count('Prof.')+maintext.count('Rev.')+maintext.count('Rt. Hon.')+maintext.count('Sr.')+maintext.count('St.')  print('Total number of sentences:',int(OrgSentence - SomeExceptions))  c = maintext.split('\n\n') #Just count lines that separates paragraphs lol  print('Total number of paragraphs:',len(c))  input()    def sub5():  # Sub-problem 5  os.system('cls') #clear screen  LeFunctionWords=['a', 'about', 'above', 'across', 'after', 'afterwards', 'again', 'against', 'all', 'almost', 'alone', 'along', 'already', 'also', 'although', 'always', 'am', 'among', 'amongst', 'amoungst', 'an', 'and', 'another', 'any', 'anyhow', 'anyone', 'anything', 'anyway', 'anywhere', 'are', 'around', 'as', 'at', 'be', 'became', 'because', 'been', 'before', 'beforehand', 'behind', 'being', 'below', 'beside', 'besides', 'between', 'beyond', 'both', 'but', 'by', 'can', 'cannot', 'could', 'dare', 'despite', 'did', 'do', 'does', 'done', 'down', 'during', 'each', 'eg', 'either', 'else', 'elsewhere', 'enough', 'etc', 'even', 'ever', 'every', 'everyone', 'everything', 'everywhere', 'except', 'few', 'first', 'for', 'former', 'formerly', 'from', 'further', 'furthermore', 'had', 'has', 'have', 'he', 'hence', 'her', 'here', 'hereabouts', 'hereafter', 'hereby', 'herein', 'hereinafter', 'heretofore', 'hereunder', 'hereupon', 'herewith', 'hers', 'herself', 'him', 'himself', 'his', 'how', 'however', 'i', 'ie', 'if', 'in', 'indeed', 'inside', 'instead', 'into', 'is', 'it', 'its', 'itself', 'last', 'latter', 'latterly', 'least', 'less', 'lot', 'lots', 'many', 'may', 'me', 'meanwhile', 'might', 'mine', 'more', 'moreover', 'most', 'mostly', 'much', 'must', 'my', 'myself', 'namely', 'near', 'need', 'neither', 'never', 'nevertheless', 'next', 'no', 'nobody', 'none', 'noone', 'nor', 'not', 'nothing', 'now', 'nowhere', 'of', 'off', 'often', 'oftentimes', 'on', 'once', 'one', 'only', 'onto', 'or', 'other', 'others', 'otherwise', 'ought', 'our', 'ours', 'ourselves', 'out', 'outside', 'over', 'per', 'perhaps', 'rather', 're', 'same', 'second', 'several', 'shall', 'she', 'should', 'since', 'so', 'some', 'somehow', 'someone', 'something', 'sometime', 'sometimes', 'somewhat', 'somewhere', 'still', 'such', 'than', 'that', 'the', 'their', 'theirs', 'them', 'themselves', 'then', 'thence', 'there', 'thereabouts', 'thereafter', 'thereby', 'therefore', 'therein', 'thereof', 'thereon', 'thereupon', 'these', 'they', 'third', 'this', 'those', 'though', 'through', 'throughout', 'thru', 'thus', 'to', 'together', 'too', 'top', 'toward', 'towards', 'under', 'until', 'up', 'upon', 'us', 'used', 'very', 'via', 'was', 'we', 'well', 'were', 'what', 'whatever', 'when', 'whence', 'whenever', 'where', 'whereafter', 'whereas', 'whereby', 'wherein', 'whereupon', 'wherever', 'whether', 'which', 'while', 'whither', 'who', 'whoever', 'whole', 'whom', 'whose', 'why', 'whyever', 'will', 'with', 'within', 'without', 'would', 'yes', 'yet', 'you', 'your', 'yours', 'yourself', 'yourselves']  dd = maintext.split()  print('Total number of function words:')  for i in range(len(LeFunctionWords)): #Guess its obvious I don't need to explain.  if dd.count(LeFunctionWords[i]) >= 1:  print(LeFunctionWords[i],dd.count(LeFunctionWords[i]))  input()    def bonus():  #Bonus  print("")  print("Made by Kasep6720")  print("")  print("Reference:")  print("UI from Shiu Shiu")  print("Some codes from bd")  print("https://www.geeksforgeeks.org/find-frequency-of-each-word-in-a-string-in-python/")  print("https://www.geeksforgeeks.org/python-program-to-calculate-the-number-of-words-and-characters-in-the-string/")  print("https://www.w3schools.com/python/python\_try\_except.asp")  print("https://www.btb.termiumplus.gc.ca/tpv2guides/guides/wrtps/index-eng.html?lang=eng&lettr=indx\_catlog\_a&page=9NBnYuQ324Yc.html")  print("")  print("Thank you!")  input()    datext = "text.txt"    if \_\_name\_\_ == "\_\_main\_\_" :  choice = ''  while (choice!='H' and choice!='h'):  os.system('cls') #clear screen  try:  f = open(datext, "r")  maintext = f.read()  except:  print("Error")  print("File not found, or language not supported")  print("Please enter another .txt file")  time.sleep(2.2)  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*YLMASS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Composition Analyzer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changing text file\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print("Current text:", datext)  datext = input("Text File Name: ")  os.system('cls') #clear screen  else:  f = open(datext, "r")  maintext = f.read()  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*YLMASS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*Composition Analyzer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*A.Total Number of words and characters \*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*B.Frequencies of letter \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*C.Frequencies of a given word \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*D.Total Number of sentences and paragraphs\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*E.Total Number of function word \*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*F.Change File \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*G.Credits \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*H.Exit \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  choice = input("Input a choice (A,B,C,D,E,F,G,H,): ")  while ((choice<'A') or (choice>'H')) and ((choice<'a') or (choice>'h')):  print("Input out of range, please input again.")  choice = input("Input a choice (A,B,C,D,E,F,G,H): ")  if (choice == 'A' or choice == 'a'):  sub1()  if (choice == 'B' or choice == 'b'):  sub2()  if (choice == 'C' or choice == 'c'):  sub3()  if (choice == 'D' or choice == 'd'):  sub4()  if (choice == 'E' or choice == 'e'):  sub5()  if (choice == 'F' or choice == 'f'):  os.system('cls') #clear screen  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*YLMASS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Composition Analyzer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changing text file\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')  print("Current text:", datext)  datext = input("Text File Name: ")  if (choice == 'G' or choice == 'g'):  bonus()  print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bye Bye!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*') |

# Acknowledgement

# Ben Shiu

# Tom Chan

# Reference

<https://www.btb.termiumplus.gc.ca/tpv2guides/guides/wrtps/index-eng.html?lang=eng&lettr=indx_catlog_a&page=9NBnYuQ324Yc.html>

<https://www.w3schools.com/python/python_try_except.asp>

<https://www.geeksforgeeks.org/python-program-to-calculate-the-number-of-words-and-characters-in-the-string/>

<https://www.geeksforgeeks.org/find-frequency-of-each-word-in-a-string-in-python/>