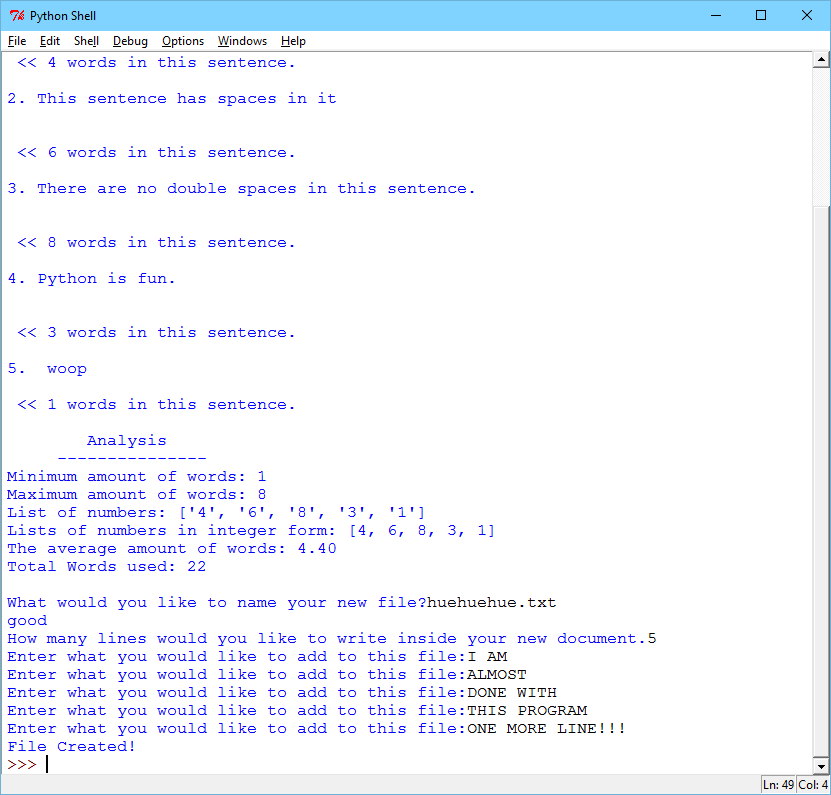
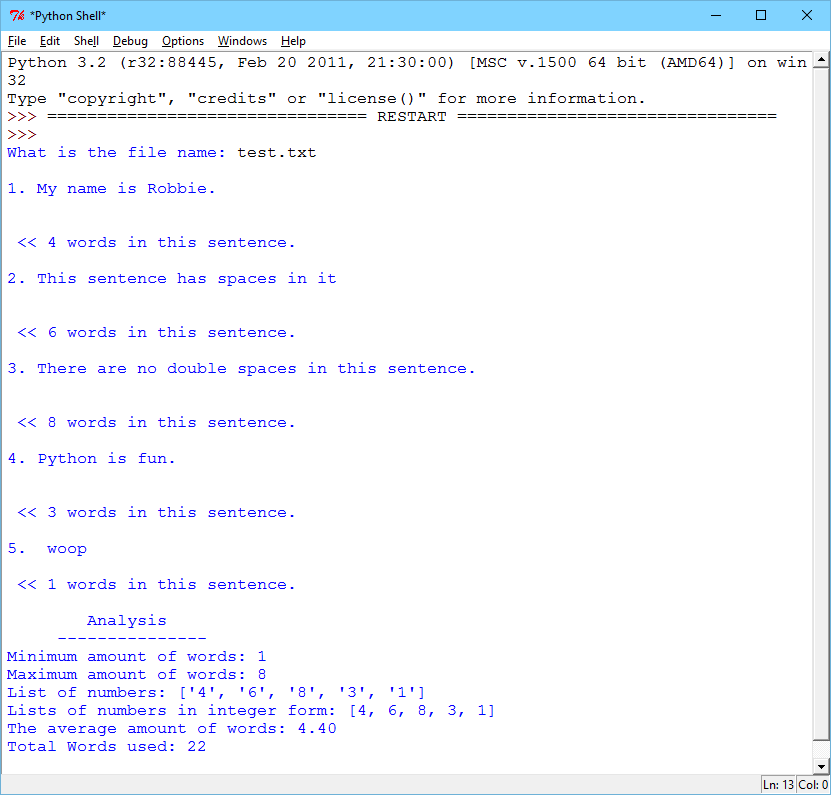
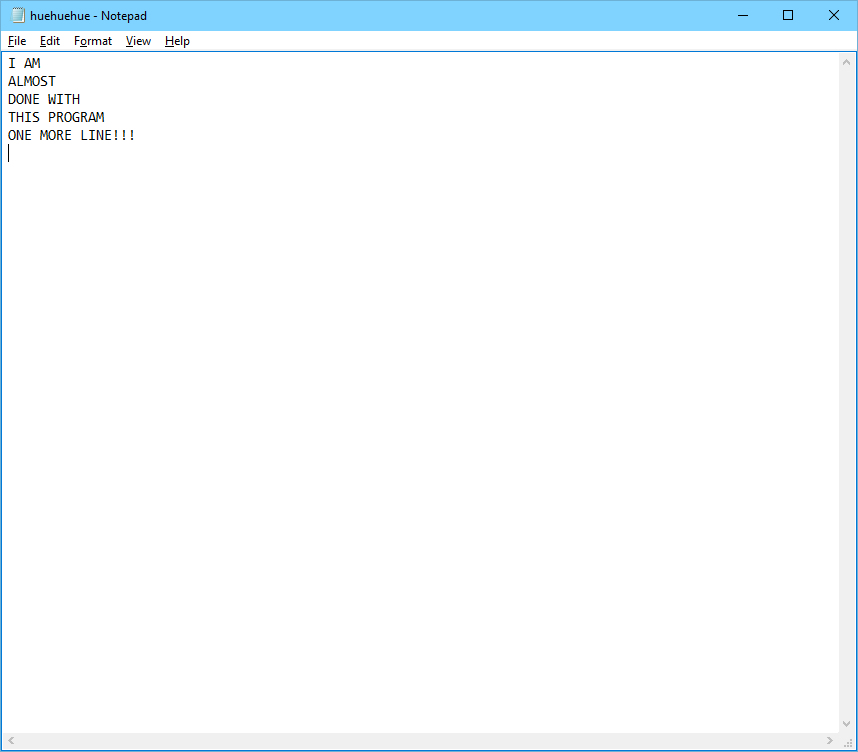
Top of Program ShellBottom of Program Shell



Txt file that was output by the program



I put a lot of effort into this program and hopefully fulfilled all of the requirements. I’m satisfied though. Enjoy! huehuehue

#Robert Cyril Plata III

#Sally Kyvernitis

#lab 7 I/O

#10/20/2015

#func reads file returns file

def file(prompt):

go = True

while go:

try:

fileName = input(prompt)

readFile = open(fileName, "r")

go = False

except:

print("'"+fileName+"'"+" does not exist")

return readFile

#func strips the line's spaces to make count accurate

def strip(sentence):

try:

hasMultiSpaces=True

while hasMultiSpaces:

oldLength = len(sentence)

sentence = sentence.replace(" "," ")

if (len(sentence) == oldLength):

hasMultiSpaces=False

except Exception as err:

print(err)

return sentence

#count counts the amount of spaces therefore counting words

def count(sentence):

wordCount=0

sentence = sentence.strip()

for i in range (0, len(sentence)):

if (sentence[i]==" "):

wordCount += 1

if (len(sentence)>0):

wordCount += 1

return wordCount

# outputs a new file into the directory of this python program

def outFile(prompt, prompt2, prompt3):

try:

go = True

while go:

userFile = input(prompt) # user makes file name. User must add .txt

if "?" in userFile:

print ("Cannot use"+"'?'")

elif ":" in userFile:

print("Cannot use"+"':'")

elif "?" and ":" not in userFile:

go = False

print ("good")

file = open(userFile, "w") # write

except Exception as err:

print (err)

tf = True

while tf:

try:

lineInput = input(prompt3) # amount of lines to be typed in the new program

lineInput = int(lineInput)

tf = False

except:

("Please Enter a number:") # enter a number

count = 0

while count < lineInput: # when count reaches line input, the writing will come to an end.

userInput = input(prompt2)

userInput =(userInput+"\n")

file.write(userInput)

count += 1

return userFile

#average of words in the file, using a list!

def avg(list):

avg = sum(myList) / float(len(myList)) \* 1

avg = avg / 1

return avg

#Main

try:

file1 = file("What is the file name: ") # name of file to open

total = 0 #not a constant

i = 1 #not a constant

myList = []

for lineOfText in file1: # reading each line of text

cleanedLine = strip(lineOfText) # removing spaces that don't need to be.

countedWords = count(cleanedLine)

print()# creating space

print(str(i)+". "+str(cleanedLine))#puts line number in

i += 1#line number added to loop each time by 1

print("\n","<< " + str(countedWords) + " words in this sentence.")

total += count(cleanedLine)

myList += (str(count(cleanedLine)))#puts number into a list each line at a time when the loop restarts.

print()

print(format("Analysis",'^25s')) #formatted title

print(format("---------------",'^25s'))#formatted dividing border

print("Minimum amount of words:", min(myList))#minimum string in list

print("Maximum amount of words:", max(myList))#maximum string in list

print ("List of numbers:",(myList))#list of string numbers

myList = [int(i) for i in myList] #makes myList a list of integers

print ("Lists of numbers in integer form:", myList)#list of integer numbers

average = avg(myList)#average using average function

print ("The average amount of words:",format(average,'4.2f'))#average number of words printed and formatted

print("Total Words used:", total)#total words used

print()

newFile = outFile("What would you like to name your new file?", "Enter what you would like to add to this file:" #3 prompts for questions in outFile() func

, "How many lines would you like to write inside your new document.")

print("File Created!")#notifying user that the program has done its job

except Exception as error: #except Exception to help me troubleshoot

print(error)