Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Guadalajara

Maestría en Ciencias de la Computación

TC4002.1 Análisis, diseño y construcción de software
Dr. Gerardo Padilla Zarate

Nombre: Héctor Gabriel Olagues Torres

Matrícula: A00354877

Febrero de 2021

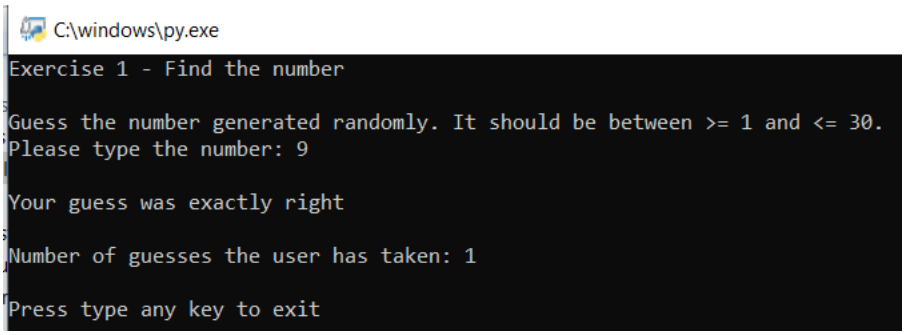
Lab 1 – Development Exercises

Programming Exercise 1 – FIND THE NUMBER

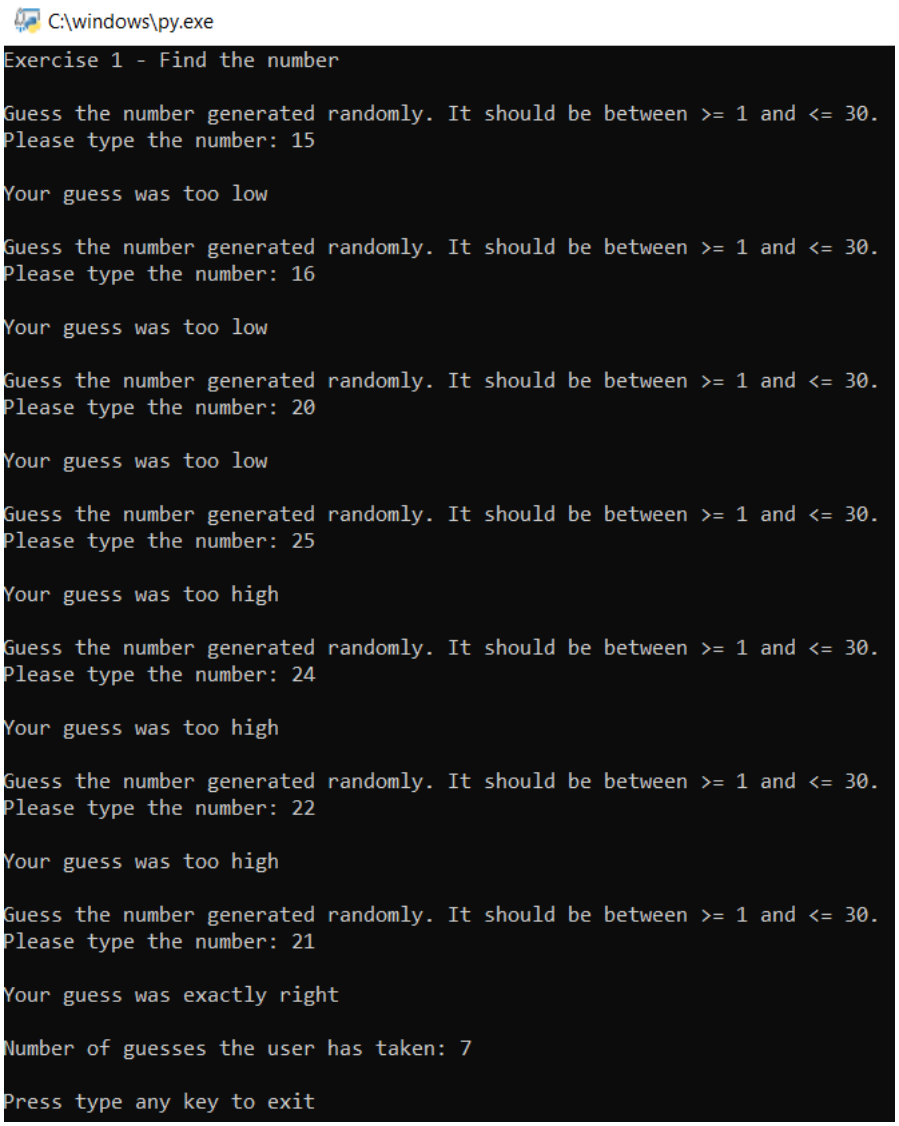
```
# Libraries
from random import randint

print("Exercise 1 - FIND THE NUMBER")

# Initialize the guesses counter
guesses = 0

# Generate random number between 1 and 30
randomNum = randint(1, 30)

inputValue = ""
inputValueInt = 0
while inputValue != "exit" and inputValueInt != randomNum:
    # Ask the user to guess the number
    print("\nGuess the number generated randomly. It should be between >= 1 and
<= 30.")
    inputValue = input("Please type the number: ")
    if inputValue != "exit":
        inputValueInt = int(inputValue)
        # Check if the number is within the accepted range
        if inputValueInt < 1:
            print("\nThe number should be >= 1.")
        elif inputValueInt > 30:
            print("\nThe number should be <= 30.")
        else:
            # Check whether the guess was too low, too high, or exactly right
            if inputValueInt > randomNum:
                print("\nYour guess was too high")
            elif inputValueInt < randomNum:
                print("\nYour guess was too low")
            else:
                print("\nYour guess was exactly right")
        # Increment the number of guesses
        guesses += 1

# Print the number of guesses the user has taken
print("\nNumber of guesses the user has taken: %d" %guesses)

# Store the number of guesses the user has taken in a file named
GuessingSteps.txt
guessesString = str(guesses)
file = open("GuessingSteps.txt", "w")
file.write(guessesString)
file.close()

input("\nPress type any key to exit")
```

Evidence

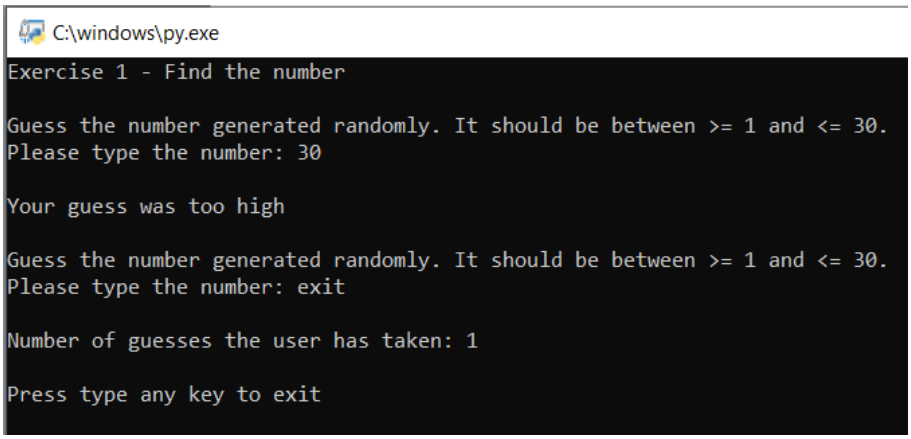
First run:



Second run:



Third run:



Programming Exercise 2 – CONVERTER

```
# This function converts from decimal format to Binary format
def decimalToBinary(numDecimal):
    if numDecimal > 1:
        # Recursive approach
        decimalToBinary(numDecimal // 2)
    print(numDecimal % 2, end="")

# This function converts from decimal format to Hexadecimal format
def decimalToHex(numDecimal):
    if numDecimal > 15:
        # Recursive approach
        decimalToHex(numDecimal // 16)
    tempVal = numDecimal % 16
    if tempVal < 10:
        print(tempVal, end="")
    if tempVal == 10:
        print("A", end="")
    if tempVal == 11:
        print("B", end="")
    if tempVal == 12:
        print("C", end="")
    if tempVal == 13:
        print("D", end="")
    if tempVal == 14:
        print("E", end="")
    if tempVal == 15:
        print("F", end="")


print("Exercise 2 - CONVERTER")

inputString = input("\nPlease type a positive integer number, including zero:
")

# Condition to check whether the input paramter contains only numeric digits
if inputString.isdecimal():
    # Only numeric digits
    inputValue = int(inputString)
    # Condition to check only positive integers are converted, including zero
    if inputValue < 0:
        print("\nOnly positive integers (plus zero) are allowed")
    else:
        # Print the numbers in Binary and Hexadecimal format
        print("\nInput value parameter in Binary: ")
        decimalToBinary(inputValue)
        print("\n\nInput value parameter in Hexadecimal: ")
        print("0x", end="")
        decimalToHex(inputValue)
else:
    # There is one or more digits that are not numeric
    print("\nThe input parameter does not contain only numeric digits")


input("\n\nPress type any key to exit")
```

Evidence

Test Case ID	Test Case Name	Input	Expected result	Result
EX2.1	Type a not numeric input parameter	When prompted to type the input parameter, provide the input with digits different to the numeric digits, e.g. “F”.	“The input parameter does not contain only numeric digits” shall be displayed and the program shall be terminated.	Expected message was displayed and the program reached the end to be terminated by typing any key.
EX2.2	Type a negative number as input parameter	When prompted to type the input parameter, provide the input with a negative number.	“The input parameter does not contain only numeric digits” shall be displayed and the program shall be terminated.	Expected message was displayed and the program reached the end to be terminated by typing any key.
EX2.3	Type a zero (“0”) as input parameter	When prompted to type the input parameter, provide the input with a zero = “0”.	The function decimalToBinary shall not be called recursively due to the input value is not greater than 1, as well as the function decimalToHex shall not be called recursively due to the input value is not greater than 15. Furthermore, “Input value parameter in Binary: 0” and “Input value parameter in Hexadecimal: 0x0” shall be displayed.	The functions decimalToBinary and decimalToHex were not called recursively and expected messages with expected conversion values were displayed.
EX2.4	Type a one (“1”) as input parameter	When prompted to type the input parameter, provide the input with a one = “1”.	The function decimalToBinary shall not be called recursively due to the input value is not greater than 1, as well as the function decimalToHex shall not be called recursively due to the input value is not greater than 15. Furthermore, “Input value parameter in Binary: 1” and “Input value parameter in Hexadecimal: 0x1” shall be displayed.	The functions decimalToBinary and decimalToHex were not called recursively and expected messages with expected conversion values were displayed.
EX2.5	Type a nine (“9”) as input parameter	When prompted to type the input parameter, provide the input with a nine = “9”.	The function decimalToHex shall not be called recursively due to the input value is not greater than 15. Furthermore, “Input value parameter in Binary: 1001” and “Input value parameter in Hexadecimal: 0x9” shall be displayed.	The function decimalToHex was not called recursively and expected messages with expected conversion

				values were displayed.
EX2.6	Type a ten (“10”) as input parameter	When prompted to type the input parameter, provide the input with a ten = “10”.	The function decimalToHex shall not be called recursively due to the input value is not greater than 15. Furthermore, “Input value parameter in Binary: 1010” and “Input value parameter in Hexadecimal: 0xA” shall be displayed.	The function decimalToHex was not called recursively and expected messages with expected conversion values were displayed.
EX2.7	Type a ten (“12”) as input parameter	When prompted to type the input parameter, provide the input with a twelve = “12”.	The function decimalToHex shall not be called recursively due to the input value is not greater than 15. Furthermore, “Input value parameter in Binary: 1100” and “Input value parameter in Hexadecimal: 0xC” shall be displayed.	The function decimalToHex was not called recursively and expected messages with expected conversion values were displayed.
EX2.8	Type a ten (“15”) as input parameter	When prompted to type the input parameter, provide the input with a fifteen = “15”.	The function decimalToHex shall not be called recursively due to the input value is not greater than 15. Furthermore, “Input value parameter in Binary: 1111” and “Input value parameter in Hexadecimal: 0xF” shall be displayed.	The function decimalToHex was not called recursively and expected messages with expected conversion values were displayed.
EX2.9	Type a sixteen (“16”) as input parameter	When prompted to type the input parameter, provide the input with a sixteen = “16”.	The function decimalToHex shall be called recursively as the input value is greater than 15. Furthermore, “Input value parameter in Binary: 10000” and “Input value parameter in Hexadecimal: 0x10” shall be displayed.	The function decimalToHex was called recursively and expected messages with expected conversion values were displayed.
EX2.10	Type following number as input parameter: 111 111 111 111	When prompted to type the input parameter, provide the input with “111 111 111 111”.	“The input parameter does not contain only numeric digits” shall be displayed and the program shall be terminated as the input parameter contains space character which is not numeric.	Expected message was displayed and the program reached the end to be terminated by typing any key.

EX2.1

```
C:\windows\py.exe
Exercise 2 - CONVERTER

Please type a positive integer number, including zero: F

The input parameter does not contain only numeric digits


Press type any key to exit
```

EX2.2

```
C:\windows\py.exe
Exercise 2 - CONVERTER

Please type a positive integer number, including zero: -5

The input parameter does not contain only numeric digits


Press type any key to exit
```

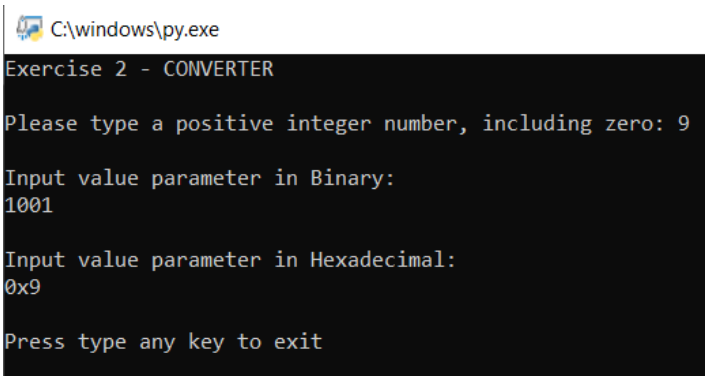
EX2.3

```
C:\windows\py.exe
Exercise 2 - CONVERTER

Please type a positive integer number, including zero: 0

Input value parameter in Binary:
0

Input value parameter in Hexadecimal:
0x0

Press type any key to exit
```

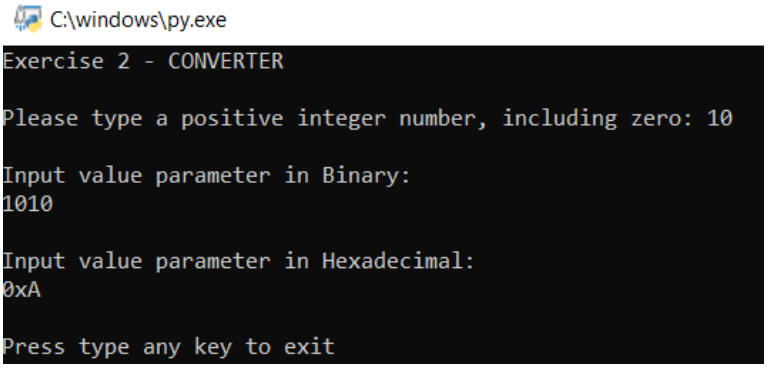
EX2.4

```
C:\windows\py.exe
Exercise 2 - CONVERTER

Please type a positive integer number, including zero: 1

Input value parameter in Binary:
1

Input value parameter in Hexadecimal:
0x1

Press type any key to exit
```

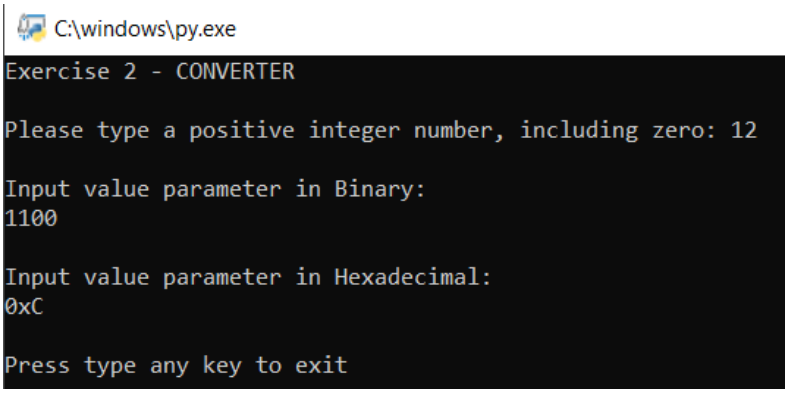
EX2.5



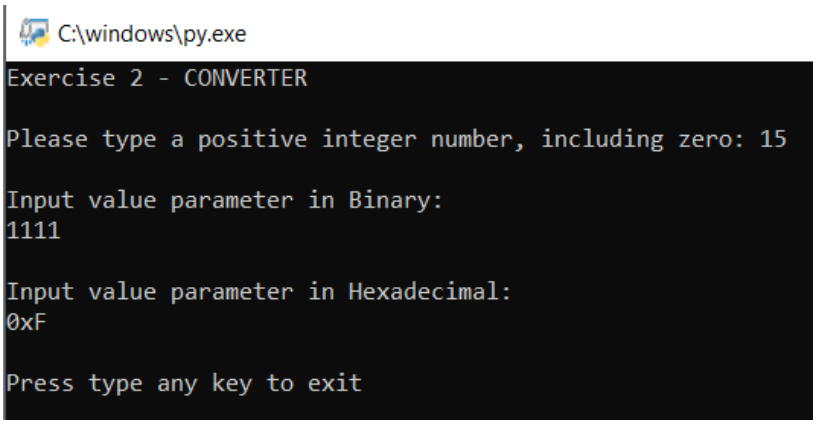
EX2.6



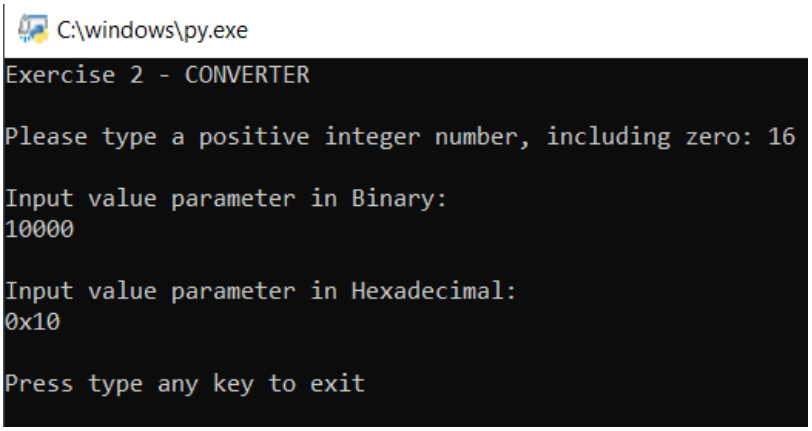
EX2.7



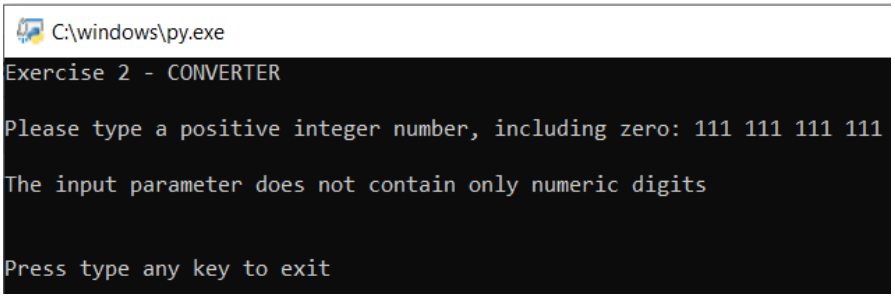
EX2.8



EX2.9



EX2.10



Programming Exercise 3 – COUNT WORDS

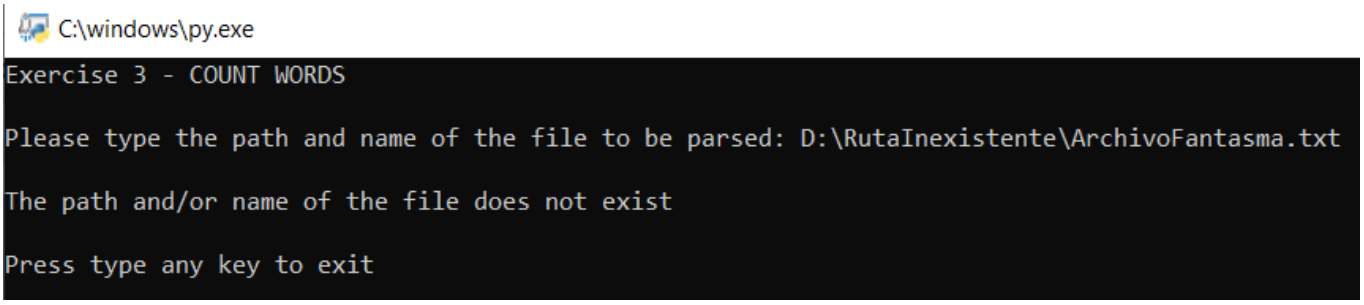
```
# Libraries
from os import path

#Functions
def countWords(fileContents, words):
    # Create a dictionary to count by word identified
    wordCoundDict = {}
    for word in words:
        wordCoundDict[word] = 0
    # Split the string of fileContents into a list
    wordsList = fileContents.split()
    # Iterate through the list of words
    for word in wordsList:
        # Increment in case a word from the list has been found in the dictionary
        if word in wordCoundDict:
            wordCoundDict[word] += 1
    # Return the dictionary to be printed along with the count of each key-value pair
    return wordCoundDict


print("Exercise 3 - COUNT WORDS")

# Input parameter indicating the path and name of the file
inputString = input("\nPlease type the path and name of the file to be parsed: ")
# Condition to check whether the path exists
if path.exists(inputString):
    # Open the file, read the contents and close it accordingly
    file = open(inputString, "r")
    fileRead = file.read()
    file.close()
    # Input parameter indicating the words to be counted
    inputWords = input("\nPlease type the words to be counted, separated by a
space:\n")
    inputWordsList = inputWords.split()
    # Call the function that actually counts the words
    countResult = countWords(fileRead, inputWordsList)
    print(countResult)
else:
    # The path or the file are incorrect
    print("\nThe path and/or name of the file does not exist")


input("\nPress type any key to exit")
```

Evidence

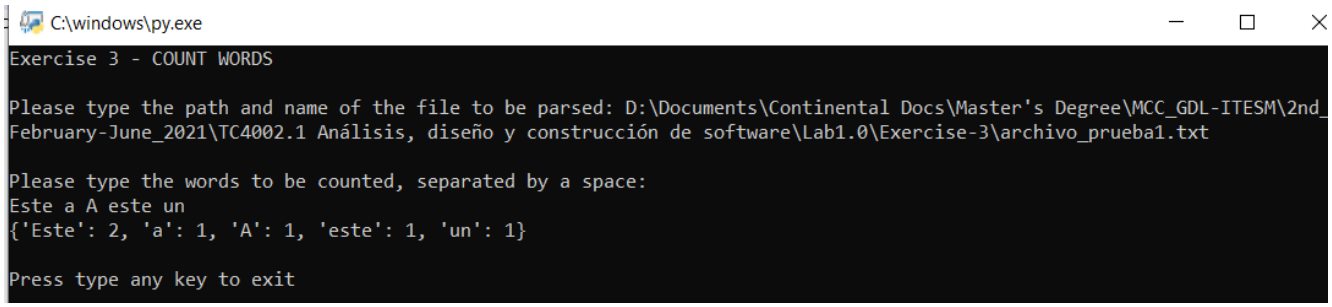
Test Case ID	Test Case Name	Input	Expected result	Result
EX3.1	Type a file or a path that does not exist	When prompted to type the path and name of the file to be	“The path and/or name of the file does not exist” shall be displayed and the program shall be terminated.	Expected message was displayed and the program reached the end

		parsed, provide an incorrect path or file name.		to be terminated by typing any key.
EX3.2	Type the path and file name correctly, as well as a list of words that do exist in such file	When prompted to type the path and name of the file to be parsed, provide a correct path and file name, as well as a list of words that do exist in the file separated by a space.	The list of words entered as parameter shall be displayed along with the number of occurrences for each word in the input file.	The list of words entered as parameter was displayed along with the correct number of occurrences for each word in the input file.
EX3.3	Type the path and file name correctly, as well as a list of words that do exist in such file plus one or more words that don't exist	When prompted to type the path and name of the file to be parsed, provide a correct path and file name, as well as a list of words that exist in the file, adding one or more words that don't exist in such file.	The list of words entered as parameter shall be displayed along with the number of occurrences for each word in the input file. The number of occurrences for those words that don't exist in the file shall be 0.	The list of words entered as parameter was displayed along with the correct number of occurrences for each word in the input file. The number of occurrences for the words that don't exist in the file was 0.

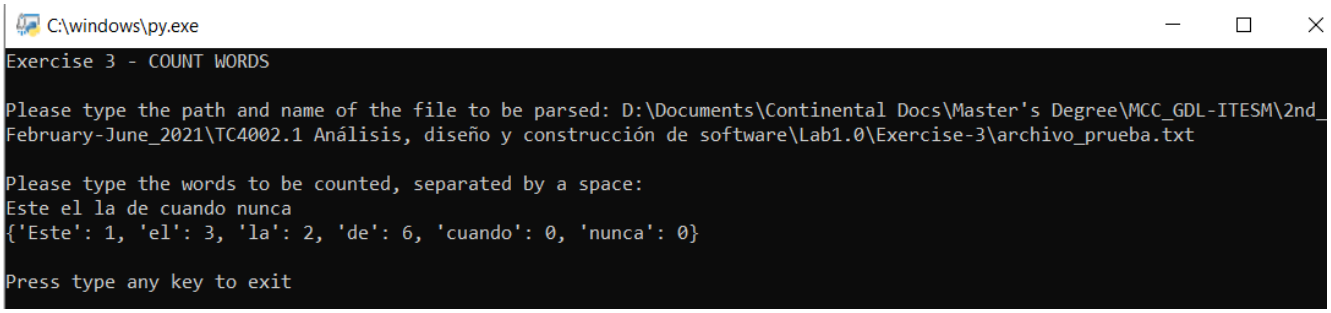
EX3.1



EX3.2


C:\windows\py.exe
Exercise 3 - COUNT WORDS
Please type the path and name of the file to be parsed: D:\Documents\Continental Docs\Master's Degree\MCC_GDL-ITESM\2nd_
February-June_2021\TC4002.1 Análisis, diseño y construcción de software\Lab1.0\Exercise-3\archivo_prueba1.txt
Please type the words to be counted, separated by a space:
Este a A este un
{'Este': 2, 'a': 1, 'A': 1, 'este': 1, 'un': 1}
Press type any key to exit

EX3.3


C:\windows\py.exe
Exercise 3 - COUNT WORDS
Please type the path and name of the file to be parsed: D:\Documents\Continental Docs\Master's Degree\MCC_GDL-ITESM\2nd_
February-June_2021\TC4002.1 Análisis, diseño y construcción de software\Lab1.0\Exercise-3\archivo_prueba.txt
Please type the words to be counted, separated by a space:
Este el la de cuando nunca
{'Este': 1, 'el': 3, 'la': 2, 'de': 6, 'cuando': 0, 'nunca': 0}
Press type any key to exit