**CSCI 1411: Fundamentals of Computing**

**Lab 11**

**Due Date: April 20, 2022**

**Name:** Malachi Milton-Brewer

**Goals:**

* Loops
* File I/O
* Error Handling

**Development Environment:** IDLE

**Deliverables:**

1. This lab handout with 6 screen shots (3 for part I and 3 for part II).
2. Your Python code for Part I of this lab. Name the file using the following format:  
   YourlastnameFirstnameLab11a.py

Example: If your name is Jamal Jones then you will name the file as follows:  
JonesJamalLab11a.py

1. Your Python code for Part II of this lab. Name the file using the following format:  
   YourlastnameFirstnameLab11b.py  
   Example: If your name is Jamal Jones then you will name the file as follows:  
   JonesJamalLab11b.py

How to take a **screen shot**:

* For a Windows 10: Use Snipping Tool to copy and CTRL + V to paste screen shot.
* For Mac: Shift + Command + 4 to copy and CTRL + V to paste screen shot.

**Part I – Skill Practice (10 Points)**

* Start IDLE
* Create a new file.
* Type the following code in the file. …..**Do not cut and paste.** You will learn more by typing it in.
* Make sure that you read all comments to understand the code
* Remember to update the first line with your own name and the date of the lab.

"""

Author: Dr. Salim Lakhani

Date: October 23, 2020

This program will perform the following tasks:

1) It will ask user for a number - number must be greater than or equal

to 2. If the user enters a number less than 2 (1, 0, or any

negative number then it will display an error message. In the same

it will display an error message if the user enters a non-numerical

or a floating point number

2) It will calculate and display all prime numbers between 2

and the user entered number.

"""

def is\_prime (num):

"""

Function Name: is\_prime

Description: Return true if the given number is prime,

otherwise it will return false. A number (n) is prime

if no number between 2 and squareroot of n can evely divide

n.

Parameter: num - an integer number

Returns true if the given number is prime, false

"""

#Calculate square root of num and convert it into int

square\_root = int(math.sqrt(num))

#Look at all numbers between 2 and square root of num

for n in range (2, square\_root+1):

#if a num can be evenly divided by n then num is not prime.

#we can use % operator. If mod is 0 then division was even

#and num is not prime

if (num % n == 0):

return False

# if all the division are not even then num is prime

return True

def main():

"""

Function Name: main

Description: Ask the user for a number greater than 2. It will

use a loop to iterate from 2 to the given number

and disply all the numbers which are prime. It will

use is\_prime function to check if the number is prime.

It will also display an error message if the input is

invalid (see program description)

Parameter: none

"""

# Set up loop control variable

error = True

#Iterate as long as error is True

# Writing *while error* is same as *while error = True*

while error:

try:

user\_input = int (input ('Enter a whole number >= 2: '))

if user\_input < 2:

print ('Input must be >= 2. Please try again')

else:

#If input > 2 then reset error to False to exit the loop

error = False

except ValueError as err:

#If input is non numerical then display an error message

print ('Input is non-numerical. Please try again')

print ('Following numbers between 2 and', user\_input, 'are prime:')

# Use a for loop to iterate through all numbers bewteen 2 and the user

# input

for num in range (2, user\_input+1):

#Use is\_prime to check if the num is prime

result = is\_prime (num)

#If result is true then display the number as prime

if result:

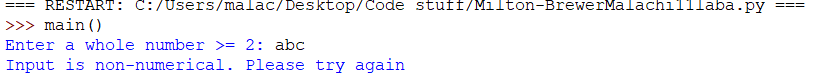
print (num, end = ' ')

* Save the file as “YourLastNameYourFirstNameLab11a.py”
* Click Run -> Run Module
* Type main() to run your program
* If there are any syntax errors then fix those errors and run your program again.
* Use the user input given in the following table to test your program for different possible outcomes.
* If you get the correct result (shown in the last column) then your program is working as expected.

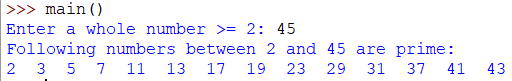
|  |  |  |
| --- | --- | --- |
| **Run Number** | **User Input** | **Output** |
| 1 | ABC | Input is non-numerical. Please try again |
| 2 | -5 | Input must be >= 2. Please try again |
| 3 | 5 | Following numbers between 2 and 5 are prime: 2 3 5 |
| 4 | 7 | Following numbers between 2 and 7 are prime: 2 3 5 7 |
| 5 | 45 | Following numbers between 2 and 45 are prime: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 |

* Once you are satisfied with your results then take a screen shot of run number 1, 2 and 5 and past them below.

**Paste your screen shot below this line**

****

****

****

**Part II – Word Counter (15 Points)**

Write a Python program that will ask user for a name of the file and a word. It will count the number of times the given word appears in the file. You must use nested loop to complete this task. The outer loop will read one line at a time from the file and inner loop will count the number of times that word appears in the line. At the end it will print the total number of times that word appear in the file. You program must also handle FileNotFoundError using try/except block. You can use the data.txt file for testing your program.

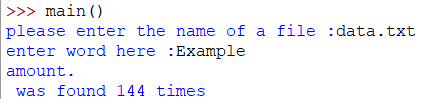
Run your program and if there are any syntax errors then fix those errors and run your program again. Use the word given in the following table to test your program for different possible outcomes. If you get the correct results (shown in the last columns) then your program is working as expected.

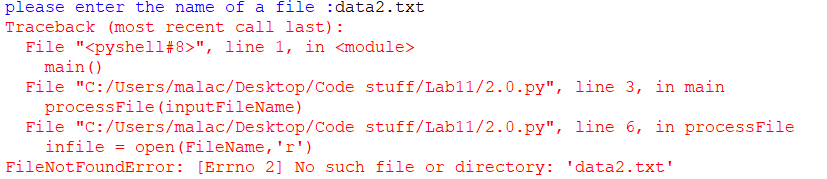
|  |  |  |  |
| --- | --- | --- | --- |
| **Run Number** | **Input** | | **Output** |
| **File Name** | **Word** |
| 1 | data2.txt |  | File data2.txt not found. Please try again |
| 2 | data.txt | This | 1 |
| 3 | data.txt | This | 2 |
| 4 | data.txt | Example: | 2 |
| 5 | data.txt | Example | 0 |
| 6 | data.txt | Jamal | 2 |
| 7 | data.txt | Discount | 4 |

**Note:** Results for run numbers 2 and 3 are different as the search is case sensitive. Also, the results for run numbers 4 and 5 are different. Why (write your answer in the following box)?

|  |
| --- |
| The code seems to read ever single word as xStr and I can’t seem to get to stop so ever output is 144. |

* Once you are satisfied with your results then take a screen shot of run number 1, 3 and 5 and past them below.





**Paste your screen shot below this line**

* Upload this lab handout with required screen shots and your code file to Canvas to submit the lab.