Assignment #1

Programming and Algorithms 2

**Due Date: 1st May 2025**

**QUESTION 1**

Review the *FULL FILE ANALYSIS* program (from Week 2) and add two options to the program:

* 1. Add an option to this program to specific the name of the file that you wish to do an analysis on.
	2. Add an option to this program to specific to number of most frequently occurring words from the file are printed out.

|  |
| --- |
| **What is the filename you wish to analyse:** ‘CompleteShakespeare.txt’**How many more frequently occurring words:** 15 |

which should return:

|  |
| --- |
| The file 'CompleteShakespeare.txt' has: 5457676 characters 124614 lines 902892 words The top 15 most occurring words are: 1. 27524 the 2. 26674 and 3. 20115 i 4. 19189 to 5. 18224 of 6. 14379 a 7. 13578 you 8. 12461 my 9. 11095 that 10. 10994 in 11. 9355 is 12. 8684 not 13. 8215 for 14. 7786 with 15. 7663 me |

**QUESTION 2**

Review the *WEB CRAWLER* program (from Week 2) and add three options to the program:

* 1. Add an option to this program to specific the web address.
	2. Add an option to this program to specific the search term.
	3. Add an option to this program to specific the maximum number of pages.

|  |
| --- |
| **What is the website you want to start with:** ‘http://www.dit.ie’**What is a search term you are looking for**: “Registration”**How many pages can be visited:** 15 |

**QUESTION 3**

Write a Python program to ask the user to type in 10 integer values, calculate the average and standard deviation of the values, write the 10 values and the average and standard deviation to a file (called Results.txt). An example of the file output is as follows:

|  |
| --- |
| 34 35 49 46 51 43 39 45 36 42 Average: 42 Standard Dev: 5.9 |

**QUESTION 4**

1. Modify the code for the “Stacks as Linked Lists” program to create a menu-driven program, with the following options:

|  |
| --- |
| * 1. Create a stack
	2. Check if the stack is full
	3. Check if the stack is empty
	4. Push an element onto the stack
	5. Pop an element off the stack
	6. Inspect the top of the stack
	7. Clear stack

99) Exit |

1. Modify the code for the “Queues as Linked Lists” program to create a menu-driven program, with the following options:

|  |
| --- |
| * 1. Create a queue
	2. Check if the queue is full
	3. Check if the queue is empty
	4. Add an element onto the queue
	5. Take an element off the queue
	6. Clear queue

99) Exit |

**QUESTION 5**

1. Add print statements to each of the sorting algorithms to show what is going on in each swap.
2. Create a single menu driven program that calls any of the 6 sorting algorithms, and asks to populate an array of up to 20 elements, and sorts it.

**Submission Instructions**

|  |
| --- |
| Please submit code, and between 200-400 words explaining your solution per part per question. Please submit in 2 ways (1) e-mail me a completed solution in a Word document., e-mail to Damian.X.Gordon@tudublin.ie with subject heading “DT249 Assignment #1”, and submit to Brightspace. |