CS2211A 2020

Jump Start Asn **05**

(optional – no credit or marks)

Assignment Five requires that node be placed at the end of the lists.
It contains a control structure that only has a pointer to the front (first node) of the list.
There are two methods to approach this assignment:
1.) figure how to add to the end with just a first (start) pointer to the first node in the list.
2.) add and 'end' pointer to the control to also hold the address of the last node in the list.

I have provided the stub of program that takes the first method.

Students have the option of completing this code as a jump start to assignment five.
This will provide experience in adding nodes to the end of a list and how to transverse the list in the order the nodes were entered. Both will be required for assignment five.

PREPARATION:
Review the class slides on Stacks and Queues.
Unzip the included C files.

Complete a C library

The goal is to create a create a linked list populated with the letters of the alphabet in order and print the list out along with the position number of each link of the list in the order they were entered.

The concepts explored are linked list, variable reference, void pointers and pointers.

You will be given most of the code in a series of C files and a header file.

You are to compete the provided code so that it produces and transverses the linked list.
Notice where in the provided code you will see:
<< ENTER YOUR CODE HERE>>

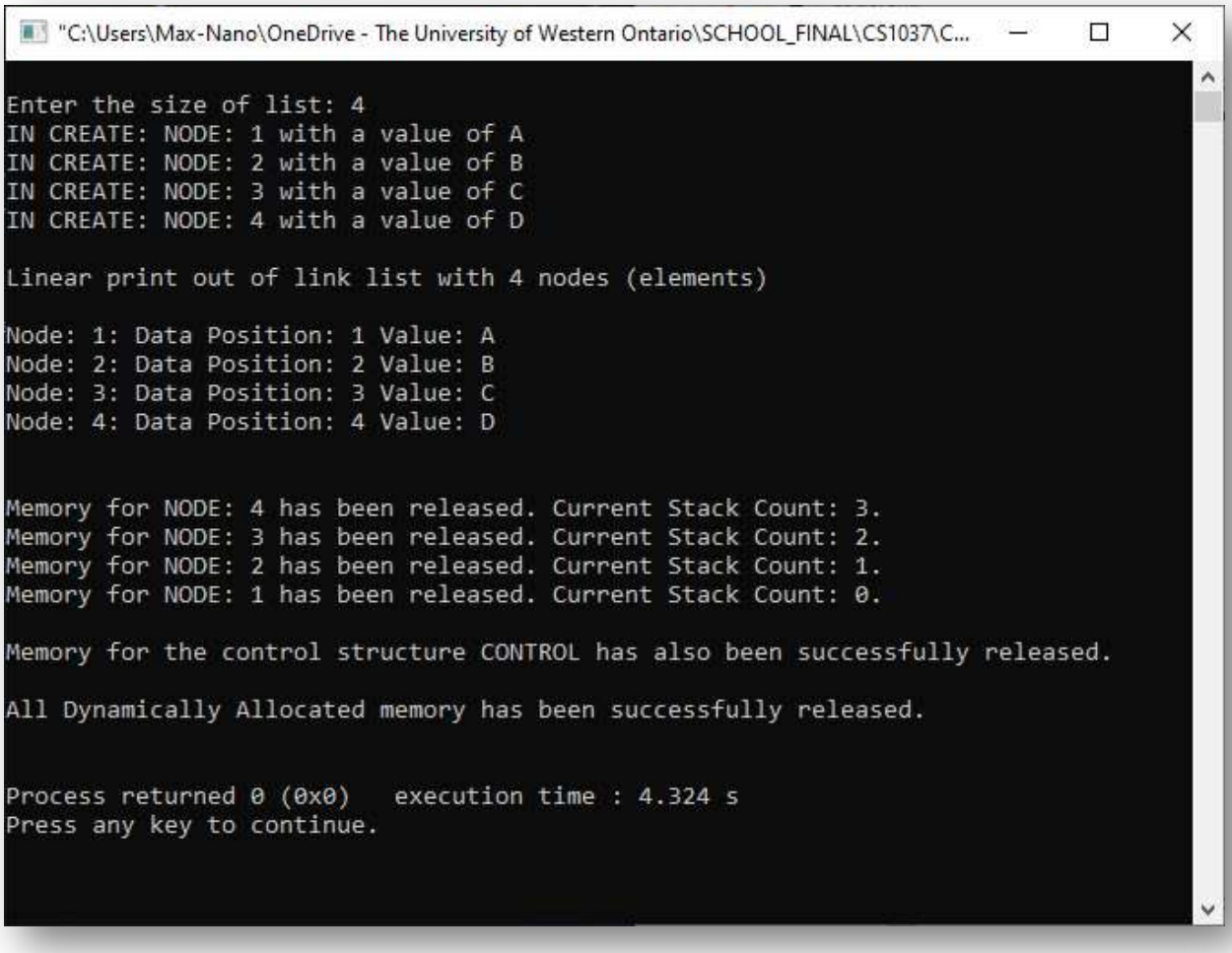
The notes and videos will be an aid in filling in the missing code.
Notice that this is neither a Stack nor a Queue, but a combination of both.

Also, this will provide an example of creating a control structure for your linked list that can serve as an example for Assignment Five (05) where you are asked to create a control structure for that linked list.

INSTRUCTIONS:
You have provided with most of the code required to create a **linked list** (a list consisting of elements that 'point' (reference) the **next** element.

Each node in the list will have a generic void pointer that in this case will point to a structure that holds and integer and a single character. One of these variables will hold a character value and the other will represent the position of when the node was created. So the first will contain 1 (representing the first node in the list), the next will contain 2, then 3, etc. representing the position of that data element when it was inserted in the list.

The program will then print out the list in the order that each node was entered.



Review and use the code in the notes as a guideline to creating your program.
The program will free the dynamic memory allocated for the list.