

Instructor: Jun Yi

Due: 10/21/2021, 11:00 pm

Utilize the doubly circular linked list that you designed in the lab to manage the data. Below are the tasks

(1)(30 points) On Friday (10/15), you designed a circular linked list. Now you must use your code. Treat it as a standard library file and call your functions in this project.

2) (40 points) In the main functions, you should insert one element into your doubly circular linked list every 0.5s. Meanwhile, you need to get one element from the linked list every 1s. As you can see, the input speed is 2 times faster than the reading speed. Thus, eventually it will be full.

The requirement is you must guarantee that you can always successfully insert the element into the linked list.

You cannot keep increasing the length of the linked list. The length should be less than 10.

3) (20 points) Once you read the element from the linked list, you should print it out to the console. You also need to print out the length of the linked list. Keep in mind that the length should be smaller than 10.

(10 points) You also need to make sure that you can stop the program. For example, if you click "enter" or whatever keys, your program will stop running.

Note:

1) The input can be whatever integers.

3) this question is to simulate one common application scenario: one client keep sending data to the other client, and the other client doesn't know how long the received data will be. In this case, you must guarantee that there is no data lost.

Hints: for the task 2, you can define a pointer which always points to the available node to insert the data. If you read a data from a node. Then this node will be available for the future insertion. If the linked list is near to full, you can read all the existing data in the linked list and print it out. Thus, you will have enough space to receive new data.