1. Name Of Project: Doubly Linked List
2. List Authors of the Project: Eli Mishriki
3. File Structure of the file submitted: This project will be submitted as a .zip file including a project DLL.cpp and a README.pdf
4. Software that can be used to run the files / Installation Guide: The compiler used to develop and test this code is Replit online IDE
5. How to get started?
    a. describe the input required from the user: When running the DLL project users will be prompted to a menu with several options that modify a prepopulated DLL. They can interact with the menu by using the numbers 1-13 on the keyboard corresponding with the menu's functions.
    b. describe what function is called by each input and briefly describe each function: Input 1 will delete the whole list and exit the menu. Input 2 calls prepend() takes the user data value and name and inserts a new node at the head of the list. Input 3 calls append() takes the user data value and name and inserts a node at the end of the list. Input 4 calls insert() takes the user data value name index and inputs a node at the index specified. Input 5 called deleteAtHead() deletes the node at the head of the list. Input 6 calls deletAtTail() and deletes the node at the tail of the list. Input 7 calls deleteAtIndex() takes a user data index and deletes the node at the given index. Input 8 calls reverseList() takes no input and returns the DLL in reverse order. Input 9 calls sortList() takes no input and returns the list in terms of values from lowest to greatest. Input 10 calls countMultiples() takes the user input value and returns the number of nodes with that value. Input 11 calls removeMultiples() takes the user input value and returns the list with all of the specified values removed. Input 12 calls headTailSplit() takes no input and returns two lists list A and list B that correspond to the first half of the DLL in list A and the second half of the DLL in reverse order in list B. Input 13 exits the menu.
6. Runtime: Derive the Big 0 of each function mathematically or by recognizing the section of code that takes the maximum time to execute: deleteAtHead() & deleteAtTail() run to O(1). deleteAtIndex() runs to O(n). sortList() uses a bubble sort which runs to O(n^2). removeMultiples() runs to O(n). countMultiples() runs to O(n). headTailSplit runs to O(n). reverseList() runs to O(n)