Lab 4 C++

Unless otherwise stated, all your programs should be created as C++ *console application*.

A deadline for demonstration and upload of code to blackboard will be announced in class

Test and Write the implementation for the following linked list code. You will receive marks for each method that is correctly implemented.

class SimpleString

{

public:

int numofCharacters;

char  \*characters;

//create memory for characters and copy the str array

SimpleString(int t\_numofCharacters,char str[]):numofCharacters(t\_numofCharacters) {

characters = new char[numofCharacters];

for (int i = 0; i<numofCharacters-1; i++)

{

characters[i] = str[i];

}

characters[numofCharacters-1] = 0;

}

SimpleString(){

numofCharacters=0;

characters=null;

}

~SimpleString();

bool isEqual(const SimpleString &d);

};

class Node

{

public:

SimpleString \*data;

Node \*next;

      //Create a SimpleString pointed to by data.

//next set to n

Node(const SimpleString &d, Node \*n = 0);

~Node();

};

class List

{

Node \*head;

public:

List();

~List();

// insert a copy of d

void insertBefore(Node \*loc, const SimpleString &d);

void insertAfter(Node \*loc, const SimpleString &d);

//push a copy of d

void push\_back(const SimpleString &d);

void push\_front(const SimpleString &d);

//returns true if  pop\_back successful. Fill d with a copy the data in the node

bool pop\_back(SimpleString &d);

//returns true if  pop\_front successful.Fill d with a copy of the data in the node

bool pop\_front(SimpleString &d);

// erase the node and delete it from memory

void erase(Node \*loc);

//Simple display the list to console using cout

void displayToConsole();

//return a pointer the the searched node. If nullptr is returned then the node is not found

Node \*search(const SimpleString &d);

};

Example:

void List::push\_front(const SimpleString &d)

{

Node \*new\_node = new Node(d, 0);

if (!head) {

head = new\_node;

return;

}

new\_node->next = head;

head = new\_node;

return;

}