**DATA LAB #7**

**DOUBLY & CIRCULAR**

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* next;

Node\* pre;

};

struct Node\* head;

void insertAtStart(int x) {

cout << "Update this function to insert " << x << "in the start of the linkedlist";

Node \* newNode = new Node();

newNode->next = NULL;

newNode->pre = NULL;

newNode->data = x;

if (head == NULL)

{

head = newNode;

return;

}

newNode->next =head;

head->pre = newNode;

head = newNode;

}

void insertAtEnd(int x) {

Node \* newNode = new Node();

Node\* temp = new Node();

newNode->next = NULL;

newNode->pre = NULL;

newNode->data = x;

if (head == NULL)

{

head = newNode;

return;

}

temp = head;

while (temp->next != NULL)

{

temp = temp->next;

}

temp->next = newNode;

newNode->pre = temp;

temp = newNode;

}

void printList() {

Node \* newNode = new Node();

Node\* temp = new Node();

temp = head;

while (temp->next != NULL)

{

cout << temp->data << "<->";

temp = temp->next;

}

cout << "NULL" << endl;

}

int main()

{

head = NULL;

insertAtStart(2);

printList();

insertAtStart(1);

printList();

insertAtEnd(3);

printList();

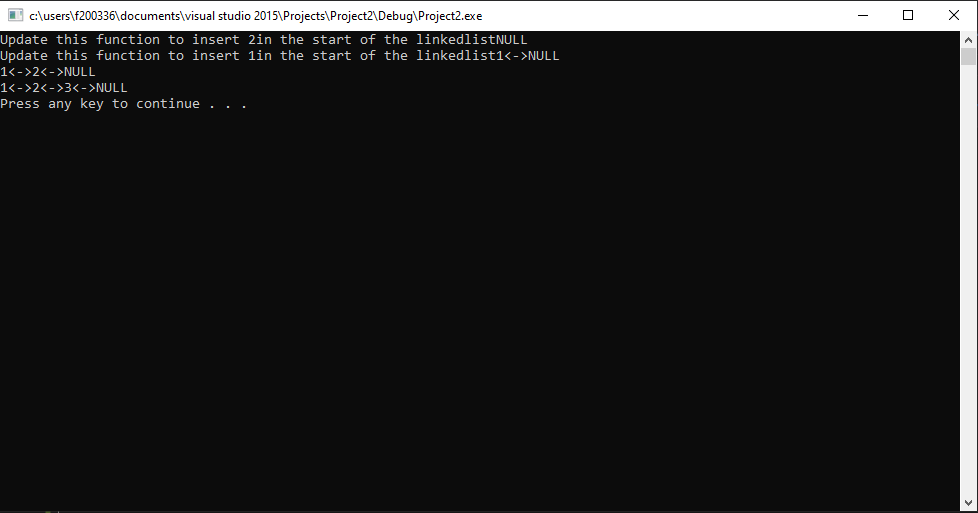
insertAtEnd(4);

printList();

system("pause");

return 0;

}

****

#include <iostream>

using namespace std;

struct Node {

int data;

Node\* link;

};

struct Node\* head;

void insertAtStart(int x) {

// cout << "Update this function to insert " << x << "in the start of the linkedlist" << endl;

Node \* newNode = new Node();

Node\* temp = head;

newNode->link = NULL;

newNode->data = x;

if (head == NULL)

{

head = newNode;

head->link = head;

return;

}

else

{

newNode->link = head;

while (temp->link != head)

{

temp = temp->link;

}

temp->link = newNode;

head = newNode;

}

}

void insertAtEnd(int x) {

// cout << "Update this function to insert " << x << "in the end of the linkedlist" << endl;

Node \* newNode = new Node();

Node \* temp = head;

newNode->link = NULL;

newNode->data = x;

if (head == NULL)

{

head = newNode;

newNode->link = head;

return;

}

else{

while (temp->link != head)

{

temp = temp->link;

}

temp->link = newNode;

newNode->link = head;

}

}

void printList() {

//cout << "Update this function to print all values in linkedlist" << endl;

Node\* temp = head;

do

{

cout << temp->data << "->";

temp = temp->link;

} while (temp->link != head);

cout << endl;

}

int main()

{

head = NULL;

insertAtStart(2);

printList();

insertAtStart(1);

printList();

insertAtEnd(3);

printList();

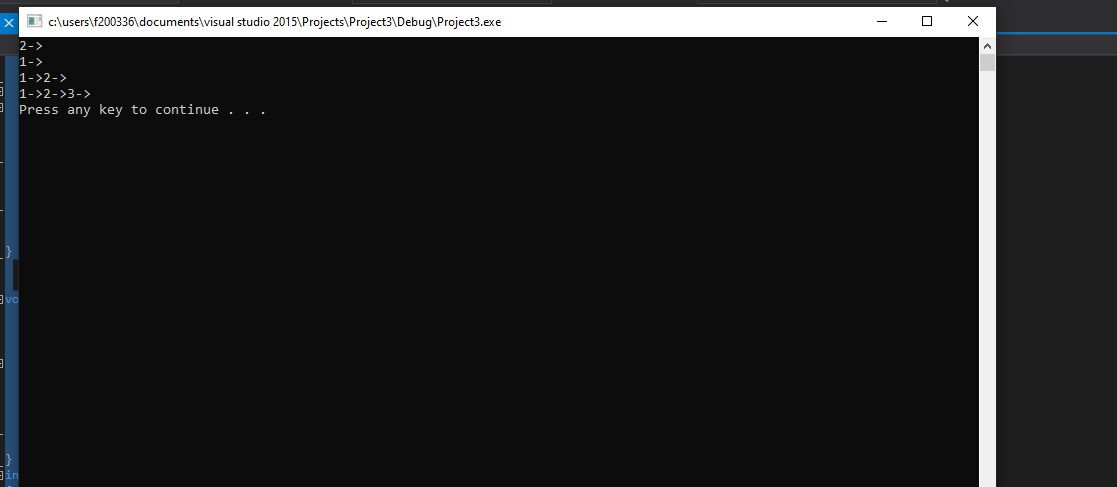
insertAtEnd(4);

printList();

system("pause");

return 0;

}

****