**Instructor: Dr. Jie Shen**

2021 Winter CIS200 – Lab 8

Release date: March. 19, 2021

Due date: March. 24, 2021

**Student Name: Demetrius Johnson**

\*Special note: see my lab uploads of the .CPP and .EXE files for ease of access and testing any of the programs for any question.

Table of Contents

[**Question 1** 3](#_Toc70436234)

[Source code (USED C++ COMPILER on Microsoft Windows 10) 3](#_Toc70436235)

[Test data and expected results 5](#_Toc70436236)

[**Question 2** 6](#_Toc70436237)

[Source code (USED C++ COMPILER on Microsoft Windows 10) 6](#_Toc70436238)

[Test data and expected results 7](#_Toc70436239)

[**Question 3** 9](#_Toc70436240)

[Source code (USED C++ COMPILER on Microsoft Windows 10) 9](#_Toc70436241)

[Test data and expected results 10](#_Toc70436242)

[**Submission** 11](#_Toc70436243)

# **Question 1**

Question: If we want to change it to a circular linked list, what should we do?

***If we want to have a circularly linked list, then we simply need to make sure that the last node (tail) in the list points to the head node of the list.***

## Source code (USED C++ COMPILER on Microsoft Windows 10)

// CIS-200-LAB\_8-DemetriusJohnson.cpp : This file contains the 'main' function. Program execution begins and ends there.

/\*

//Author: Demetrius E Johnson

//Date: April 26, 2021

//Last Modification Date: April 26, 2021

//Purpose: This program demonstrates the use of a linked list using a struct and some basic functions

\*/

/\*

Question 1:

//Write a function called listsize that takes a pointer to the start of a linked list and returns the number of elements in the list,

//and another function called listsum that also takes a pointer to the start of a linked list and returns the sum of the values of all elements in the list.

//In your main routine, you need to print out the results of these two functions on computer screen.

\*/

#include <iostream>

//#include<assert.h>

using namespace std;

//FUNCTION DECLARATIONS

struct listrec

{

int value;

struct listrec\* next;

};

int listsize(listrec\* LinkedList\_Ptr); // takes a pointer to the start of a linked list and returns the number of elements in the list

int listsum(listrec\* LinkedList\_Ptr); //takes a pointer to the start of a linked list and returns the sum of the values of all elements in the list

//FUNCTION DECLARATIONS

int main()

{

cout << "---Welcome: This program demonstrates the use of a linked list using a struct and some basic functions\n--By Demetrius Johnson\n\n";

listrec int\_LinkedList; //create a linked list; this node will keep the starting address of the list as well

listrec\* int\_linkedList\_Ptr = &int\_LinkedList; //use this ptr to navigate through the linked list

int\_linkedList\_Ptr->value = 4; //set value of FIRST node

int\_linkedList\_Ptr->next = new listrec; //allocate memory for next node

int\_linkedList\_Ptr = int\_linkedList\_Ptr->next; //move to next node

int\_linkedList\_Ptr->value = 5; //set value of SECOND node

int\_linkedList\_Ptr->next = new listrec; //allocate memory for next node

int\_linkedList\_Ptr = int\_linkedList\_Ptr->next; //move to next node

int\_linkedList\_Ptr->value = 3; //set value of THIRD node

int\_linkedList\_Ptr->next = nullptr; //we have inserted all items; set end of list to nullptr so we have a denotation for the end.

cout << "The Number of elements in the int linked list is: " << listsize(&int\_LinkedList) << endl << endl;

cout << "The sum of all the values stored by all the elements in the int linked list is: " << listsum(&int\_LinkedList) << endl << endl;

cout << endl << endl << "The program has finished execution....now exiting...thank you....\n\n";

system("pause");

return 0;

}

//FUNCTION DEFINITIONS BELOW THIS LINE

int listsize(listrec\* LinkedList\_Ptr) {

int element\_counter = 0; //use this to keep track of the size of the linked list

while (LinkedList\_Ptr != nullptr) {

element\_counter++; // current node not null; so increase counter to add an element

LinkedList\_Ptr = LinkedList\_Ptr->next; //move to next node

}

return element\_counter; //return the number of elements

} // takes a pointer to the start of a linked list and returns the number of elements in the list

int listsum(listrec\* LinkedList\_Ptr) {

int sum = 0; //use this to keep track of the sum of the element values in the linked list

while (LinkedList\_Ptr != nullptr) {

sum += LinkedList\_Ptr->value; // current node not null; so add value of the current element

LinkedList\_Ptr = LinkedList\_Ptr->next; //move to next node

}

return sum; //return sum of all the elements

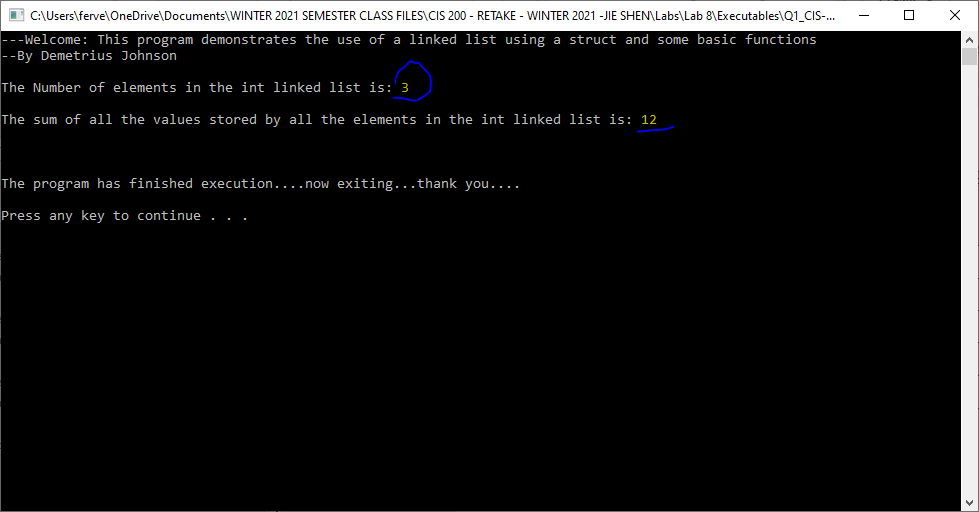
} //takes a pointer to the start of a linked list and returns the sum of the values of all elements in the list

## Test data and expected results

Test Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | valid | Call functions that tell size of the linked list and the sum of the values stored by each element | Called listsize and listsum | 3 elements, and the sum of the 3 elements should be 12 | See screenshot | pass |

TEST 1:



# **Question 2**

Question: If we want to change to a doubly-linked list, what should we do?

*For a doubly linked list, you simply have two pointers for each node: one for the next node, and one for the previous node; in this way the list is doubly linked and you can move forward and backward in the list.*

## Source code (USED C++ COMPILER on Microsoft Windows 10)

// CIS-200-LAB\_8-DemetriusJohnson.cpp : This file contains the 'main' function. Program execution begins and ends there.

/\*

//Author: Demetrius E Johnson

//Date: April 26, 2021

//Last Modification Date: April 26, 2021

//Purpose: This program demonstrates the use of a linked list using a struct and some basic functions

\*/

/\*

Question 2:

//Write a function called listsize that takes a pocharer to the start of a linked list and returns the number of elements in the list,

//and another function called listsum that also takes a pocharer to the start of a linked list and returns the sum of the values of all elements in the list.

//In your main routine, you need to prchar out the results of these two functions on computer screen.

\*/

#include <iostream>

//#include<assert.h>

using namespace std;

//FUNCTION DECLARATIONS

struct listrec

{

char value;

struct listrec\* next;

};

void printlist(listrec\* start\_of\_linked\_list); // print out all the nodes in the list

//FUNCTION DECLARATIONS

int main()

{

cout << "---Welcome: This program demonstrates the use of a linked list using a struct and some basic functions\n--By Demetrius Johnson\n\n";

listrec char\_LinkedList; //create a linked list; this node will keep the starting address of the list as well

listrec\* char\_linkedList\_Ptr = &char\_LinkedList; //use this ptr to navigate through the linked list

char\_linkedList\_Ptr->value = 'a'; //set value of FIRST node

char\_linkedList\_Ptr->next = new listrec; //allocate memory for next node

char\_linkedList\_Ptr = char\_linkedList\_Ptr->next; //move to next node

char\_linkedList\_Ptr->value = 'c'; //set value of SECOND node

char\_linkedList\_Ptr->next = new listrec; //allocate memory for next node

char\_linkedList\_Ptr = char\_linkedList\_Ptr->next; //move to next node

char\_linkedList\_Ptr->value = 'W'; //set value of THIRD node

char\_linkedList\_Ptr->next = nullptr; //we have inserted all items; set end of list to nullptr so we have a denotation for the end.

cout << "The values stored in the char linked list are: \n\n";

printlist(&char\_LinkedList);

cout << endl << endl << "The program has finished execution....now exiting...thank you....\n\n";

system("pause");

return 0;

}

//FUNCTION DEFINITIONS BELOW THIS LINE

void printlist(listrec\* LinkedList\_Ptr) {

while (LinkedList\_Ptr != nullptr) {

cout << LinkedList\_Ptr->value << " "; //cout value of current node

LinkedList\_Ptr = LinkedList\_Ptr->next; //move to next node

}

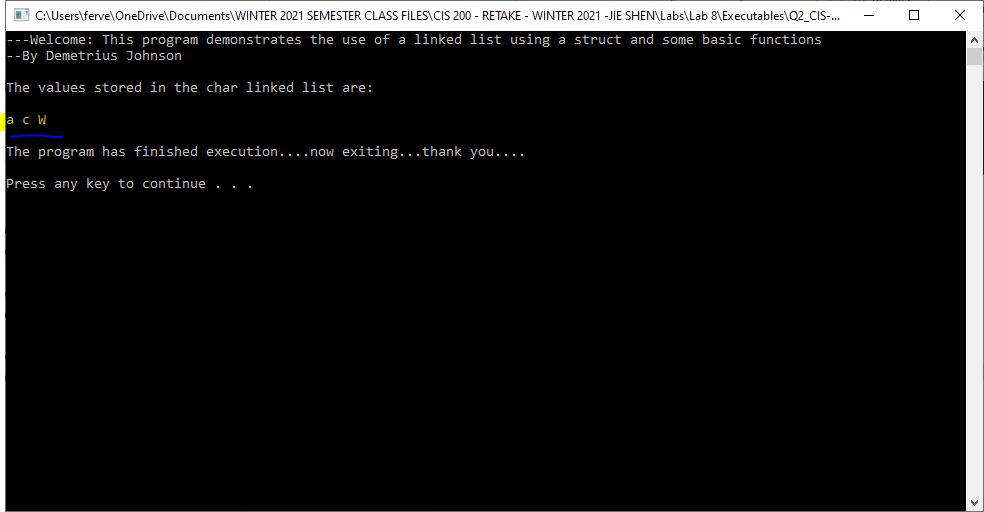
} // print out all the nodes in the list

## Test data and expected results

Test Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | valid | Print out values in the list | a, c, and W; call print function for the list | a c W | See screenshot | pass |

TEST 1:



# **Question 3**

## Source code (USED C++ COMPILER on Microsoft Windows 10)

// CIS-200-LAB\_8-DemetriusJohnson.cpp : This file contains the 'main' function. Program execution begins and ends there.

/\*

//Author: Demetrius E Johnson

//Date: April 27, 2021

//Last Modification Date: April 27, 2021

//Purpose: This program demonstrates how to output beginning memory addresses

\*/

/\*

Question 2:

//In the following main( ) function, try to find a way to print out the beginning memory address of each variable.

\*/

#include <iostream>

//#include<assert.h>

using namespace std;

int add(int x, int y)

{

return (x + y);

}

int main()

{

cout << "---Welcome: This program demonstrates how to output beginning memory addresses\n--By Demetrius Johnson\n\n";

int a = 10;

float b = 3.14;

char c = 'j';

char d[80];

// print out the beginning memory address of the above 4 variables

// as well as the memory address of function add( )

cout << "The beggning memory address of an array or function is its name;\nfor normal variables, the beginning address is simply the name using the & operator.\n";

cout << "A sepcial case for normal variables or for an array, is when you have a char variable or char array,\nthe &operator or ";

cout << "name of char array fails so you must cast it as a (void\*) pointer" << endl;

cout << "and output the value stored by the casted pointer that is storing the address\nof the char or start address of the char array.\n\n";

cout << "For example: 'add' is a function name, 'a' is an int, 'b' is a float, 'c' is a char, and 'd' is a char array, thus:\n\n";

cout << "cout << add --> " << add << endl;

cout << "cout << &a --> " << &a << endl;

cout << "cout << &b --> " << &b << endl;

cout << "cout << (void\*)&c --> " << (void\*)&c << endl;

cout << "cout << (void\*)d --> " << (void\*)d << endl;

cout << endl << endl << "The program has finished execution....now exiting...thank you....\n\n";

system("pause");

return 0;

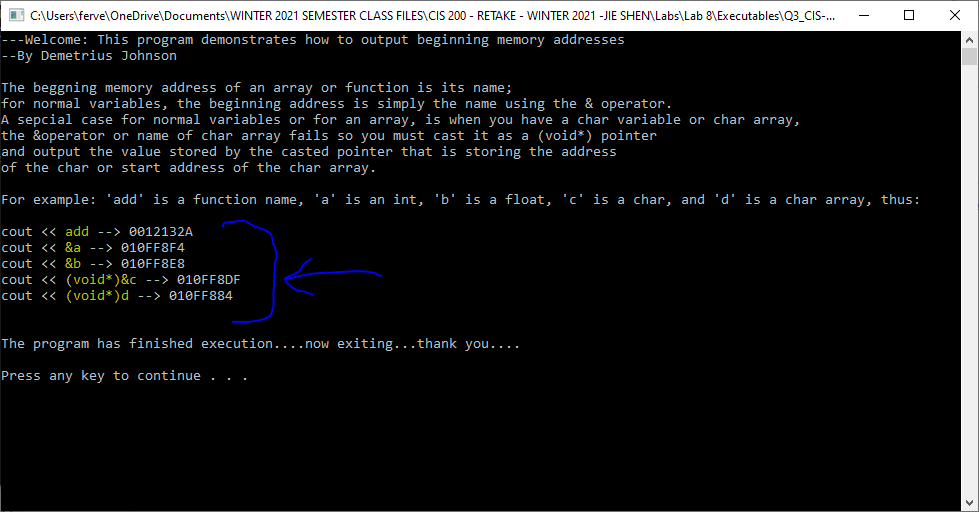
}

## Test data and expected results

Test Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | valid | Output the addresses of the different data types and function | See screenshot | See screenshot | See screenshot | pass |

TEST 1:



# **Submission**

(1) The Word document should contain the following information

* Your name
* Machine type (Unix, Mac, Linux or PC machine ?)
* Compiler type
* Description of your code design and implementation
* Inclusion of your source
* A reasonable number of comment lines in your source code
* Screen shot of your test run