**University of Michigan – Dearborn**

**CIS 200 – Computer Science 2**

**Lab10**

Nahrin Sharna

[nsharna@umich.edu](mailto:nsharna@umich.edu)

Date: 03/31/2019

# Source Code:

/\*

Author: Nahrin Sharna

Creation Date: 03/27/2019

Modification Date: 03/31/2019

Purpose: Create an unsorted linked list from file input and then sort the linked list

\*/

#include<iostream>

#include<fstream>

using namespace std;

struct node {

int num;

node \*next;

};

void mergeSort(node \*&);

node\* merge(node \*&, node \*&);

void createList(int , node \*&);

int getLength(node \*);

void printLinkedList(node \*);

int main() {

int numbers;

int count = 0;

node \*head = NULL;

ifstream ins;

ins.open("data.txt");

if (!ins.is\_open()) {

cout << "File does not exits" << endl;

}

else if (ins.peek() == EOF) {

cout << "File is empty." << endl;

}

else {

while (!ins.eof() && count != 21) {

{

ins >> numbers;

if (numbers >= -250 && numbers <= 250) {

count++;

createList(numbers, head);

}

else {

cout << numbers << " is an invalid number." << endl;

}

}

}

node \*d = head;

cout << "Unsorted List: " << endl;

do {

cout << d->num << " "; //printing unsorted list

d = d->next;

} while (d != NULL);

cout << endl;

mergeSort(head);

cout << "Sorted list: " << endl;

printLinkedList(head);

}

cout << endl;

ins.close();

system("pause");

return 0;

}

/\*

Author: Nahrin Sharna

Creation Date: 03/27/2019

Modification Date: 03/27/2019

Purpose: create the linked list with file input

\*/

void createList(int x, node \*&head) {

node \*temp = new node;

temp->num = x;

temp->next = NULL;

node \*p = head;

if (head == NULL) {

head = temp;

p = head;

}

else {

temp->next = head;

head = temp;

p = temp;

}

}

/\*

Author: Nahrin Sharna

Creation Date: 03/27/2019

Modification Date: 03/27/2019

Purpose: Return the size or length of the current linked list

\*/

int getLength(node \*head) {

node\* cur = head;

int i = 0;

for (i; cur != NULL; cur = cur->next) {

i++;

}

return i;

}

/\*

Author: Nahrin Sharna

Creation Date: 03/27/2019

Modification Date: 03/27/2019

Purpose: Sort the list

\*/

void mergeSort(node \*&head) {

if (head->next != NULL)

{

node\* head1;

node\* head2 = head;

int len = getLength(head);

head1 = head;

for (int i = 0; i < len / 2; i++)

{

head1 = head2;

head2 = head2->next;

}

head1->next = NULL;

head1 = head;

mergeSort(head1);

mergeSort(head2);

head = merge(head1, head2);

}

}

/\*

Author: Nahrin Sharna

Creation Date: 03/27/2019

Modification Date: 03/31/2019

Purpose: Merge the list

\*/

node\* merge(node \*&head1, node \*&head2) {

node\* newHead;

//Base case: return the other half if one of them is NULL

if (head1 == NULL) return head2;

else if (head2 == NULL) return head1;

//compare the value

if (head1->num < head2->num) {

newHead = head1;

newHead->next = merge(head1->next, head2);

}

else {

newHead = head2;

newHead->next = merge(head1, head2->next);

}

return newHead;

}

/\*

Author: Nahrin Sharna

Creation Date: 03/27/2019

Modification Date: 03/31/2019

Purpose: Print the list

\*/

void printLinkedList(node \*head) {

node\* cur = head;

for (; cur != NULL; cur = cur->next) {

cout << cur->num << " ";

}

}

# Initial Test Plan:

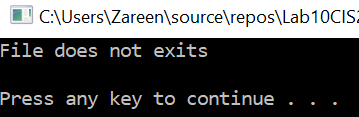
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | Invalid | When the file does not exist | filename = “data1.dat” | File does not exist. |  |  |
| 2 | Invalid | When the file is empty | Filename = “data.dat” | File is empty |  |  |
| 3 | Valid | File contains 21 random numbers between -250 to 250 | Filename = “data.dat”  Numbers: -15 6 8 30 -180 55 99 250 -33 10 42 99 -77 -99 22 150 222 -83 199 47 66 | Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 250 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 250 |  |  |
| 4 | Invalid | File contains numbers that is out of valid range. List will not contain the invalid number | Filename = “data.dat”  -15 6 8 30 -180 55 99 253 -33 10 42 99 -77 -99 22 150 222 -83 199 47 66 | 253 is an invalid number.  Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 |  |  |
| 5 | Invalid | File contains more than 21 random numbers.  Program will only take the first 21 numbers and truncate the rest. Then it will display the unsorted and sorted list containing first 21 numbers. | Filename = “data.dat”  -15 6 8 30 -180 55 99 250 -33 10 42 99 -77 -99 22 150 222 -83 199 47 66 79  Total = 22 number; 79 is the 22nd number | Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 250 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 250 |  |  |

# Final Test Plan:

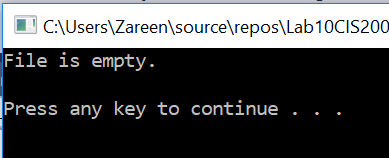
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test # | Valid / Invalid Data | Description of test | Input Value | Expected Output | Actual Output | Test Pass / Fail |
| 1 | Invalid | When the file does not exist | filename = “data1.dat” | File does not exist. | File does not exist. | Pass |
| 2 | Invalid | When the file is empty | Filename = “data.dat” | File is empty | File is empty | Pass |
| 3 | Valid | File contains 21 random numbers between -250 to 250 | Filename = “data.dat”  Numbers: -15 6 8 30 -180 55 99 250 -33 10 42 99 -77 -99 22 150 222 -83 199 47 66 | Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 250 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 250 | Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 250 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 250 | Pass |
| 4 | Invalid | File contains numbers that is out of valid range. List will not contain the invalid number | Filename = “data.dat”  -15 6 8 30 -180 55 99 253 -33 10 42 99 -77 -99 22 150 222 -83 199 47 66 | 253 is an invalid number.  Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 | 253 is an invalid number.  Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 | Pass |
| 5 | Invalid | File contains more than 21 random numbers.  Program will only take the first 21 numbers and truncate the rest. Then it will display the unsorted and sorted list containing first 21 numbers. | Filename = “data.dat”  -15 6 8 30 -180 55 99 250 -33 10 42 99 -77 -99 22 150 222 -83 199 47 66 79  Total = 22 number; 79 is the 22nd number | Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 250 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 250 | Unsorted List:  66 47 199 -83 222 150 22 -99 -77 99 42 10 -33 250 99 55 -180 30 8 6 -15  Sorted list:  -180 -99 -83 -77 -33 -15 6 8 10 22 30 42 47 55 66 99 99 150 199 222 250 | Pass |

# Screen Shots:

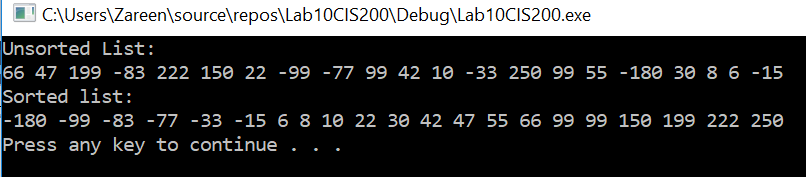
## Test Case 1:



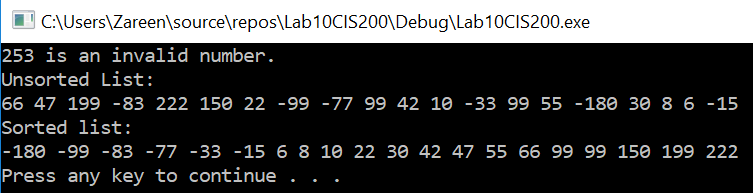
## Test Case 2:



## Test Case 3:



## Test Case 4:



## Test Case 5:

