الجامعة الاسلامية العالمية
International Islamic University, Islamabad

International Islamic University, Islamabad

Assignment #4

Object Oriented Pradigm

Linked List

Muhammad Sharjeel Husnain(4345)

Object Oriented Pradigm

Prof. Nadeem

Submission Date

December 12, 2022

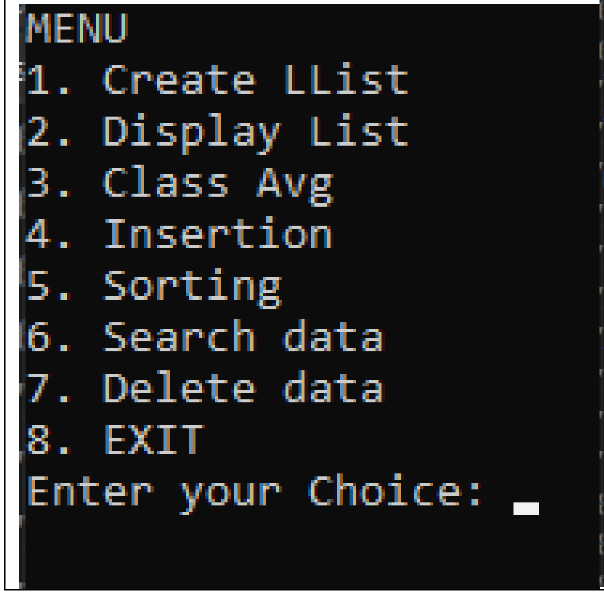
```
//Program to insert,display,search,delete a node in single Linked list
//*****Muhammad Sharjeel Husnain*****
//compilation date 03/12/2022
//Compiler Used: Microsoft visual studio(Community Eddition)

#include <iostream>
using namespace std;

class node {
	int data;
	node* next;
public:
	node() {
		data = 0;
		next = NULL;
	}

	void createLL(node*&, node*&, int);
	void displayLL(node*&);
	void averageLL(node*&);
	void insertLL(node*&,node*&);
	void sortLL(node*&, char);
	int search(node*&, int);
	void deleteLL(node*&, int);


};


void node::createLL(node*& first, node*& temp, int no)
{
	node* pre = NULL;
	for (int i = 0; i < no; i++) {
		temp = new node;
		temp->next = NULL;
		cout << "Enter Value" << endl;
		cin >> temp->data;

		if (first == NULL)
			first = pre = temp;
		else
		{
			pre->next = temp;
			pre = temp;

		}
	}
}

void node::displayLL(node*& temp) {

	while (temp != NULL)
	{
		cout << "|" << temp->data << "|\t";
```

```
			temp = temp->next;
		}
}
void node:: averageLL(node*& temp)
{
	int sum = 0, n = 0;
	float avg;

	while (temp != NULL)
	{
		sum += temp->data;
		n++;
		temp = temp->next;
	}
	avg = static_cast<float>(sum) / n;

	cout << avg << endl;
}
void node:: insertLL(node*& first, node*& temp) {

	int n = 1;
	int pos = 0;
	temp = first;
	node* ntemp = NULL;
	cout << "Enter Position where you want to insert Data" << endl;
	cin >> pos;
	while (temp != NULL)
	{
		n++;
		temp = temp->next;
	}
	if (pos >= n)
	{
		cout << "write position less then " << n << endl;

	}
	else {
		temp = new node;
		temp->next = NULL;
		cout << "Enter Data" << endl;
		cin >> temp->data;
		ntemp = new node;
		ntemp = first;
		for (int i = 0; i < pos - 1; i++)
		{
			ntemp = ntemp->next;
		}
		temp->next = ntemp->next;
		ntemp->next = temp;
	}
}
void node::sortLL(node*& first, char ch)
{
	node* a, * b;
	int ptr;
	cout << "1, sort in assending order\n"
		<< "2. sort in dessending order" << endl;
	cin >> ch;
```

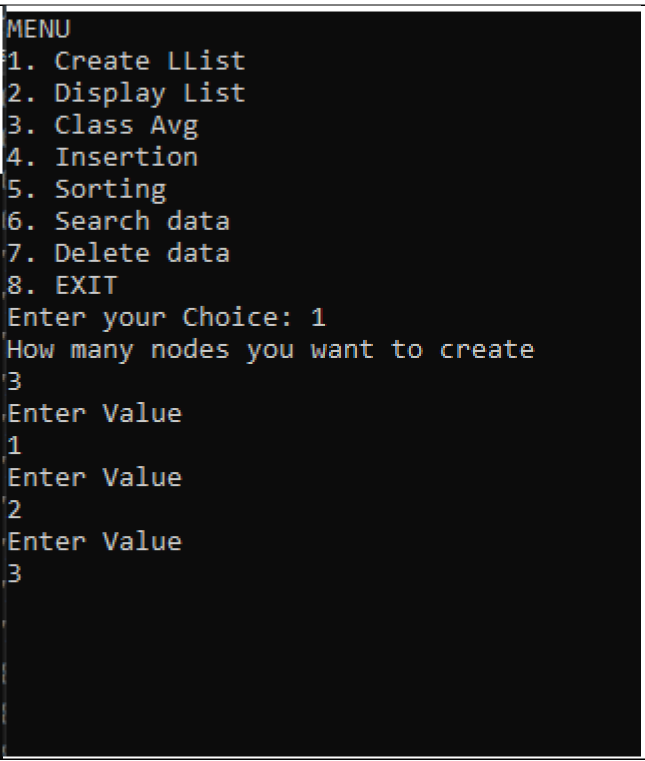
```
        if (first == NULL)
                cout << "Empty Link List" << endl;
        else
        {
                for (a = first; a->next != NULL; a = a->next)
                {
                        for (b = a->next; b != NULL; b = b->next)
                        {
                                if (ch == '1')
                                {
                                        if (a->data > b->data)
                                        {
                                                ptr = a->data;
                                                a->data = b->data;
                                                b->data = ptr;
                                        }
                                }if (ch == '2')
                                {
                                        if (b->data > a->data)
                                        {
                                                ptr = a->data;
                                                a->data = b->data;
                                                b->data = ptr;
                                        }
                                }
                        }
                }if (ch == '1')
                        cout << "sorted in assending order" << endl;
                if (ch == '2')
                        cout << "sorted in desending order" << endl;
        }
}
int node::search(node*& temp, int srch)
{
        for (int i = 0; temp != NULL; i++)
        {
                if (temp->data == srch)
                        return i;
                temp = temp->next;
        }
        return -1;
}
void node:: deleteLL(node*& first, int no)
{
        node* temp = new node;
        temp = first;
        if (no == 1)
        {
                first = temp->next;
                delete temp;
                cout << "node deleted from " << no << "Position" << endl;
        }
        else {
                for (int i = 0;i < no - 1;i++)
                {
                        temp = temp->next;
                }
                node* ntemp = temp->next;
```

```
              temp->next = ntemp->next;
              cout << "node deleted from " << no << "Position" << endl;
              delete ntemp;
       }
}

int main() {
       node list;
       node* first = NULL,

              * temp = NULL;

       int no = 0;

       int srch;
       char ch;
       do {
              cout << "MENU" << endl
                     << "1. Create LList\n"
                     << "2. Display List\n"
                     << "3. Class Avg\n"
                     << "4. Insertion\n"
                     << "5. Sorting\n"
                     << "6. Search data\n"
                     << "7. Delete data\n"
                     << "8. EXIT \n";
              cout << "Enter your Choice: ";
              cin >> ch;
              switch (ch) {
              case '1':   //create linklist
                     cout << "How many nodes you want to create" << endl;
                     cin >> no;
                     list.createLL(first, temp, no);
                     break;
              case '2':
                     temp = first;
                     list.displayLL(temp);

                     break;
              case '3':// class avg
                     temp = first;
                     list.averageLL(temp);
                     break;
              case'4'://inserting
                     list.insertLL(first, temp);
                     break;
              case '5': //sorting
                     list.sortLL(first, ch);
                     break;
              case '6': // searching
              {
                     cout << "Enter Value to Search" << endl;
                     cin >> srch;
                     temp = first;
                     int result;
                     result = list.search(temp, srch);
                     if (result == -1)
                            cout << "not found" << endl;
```

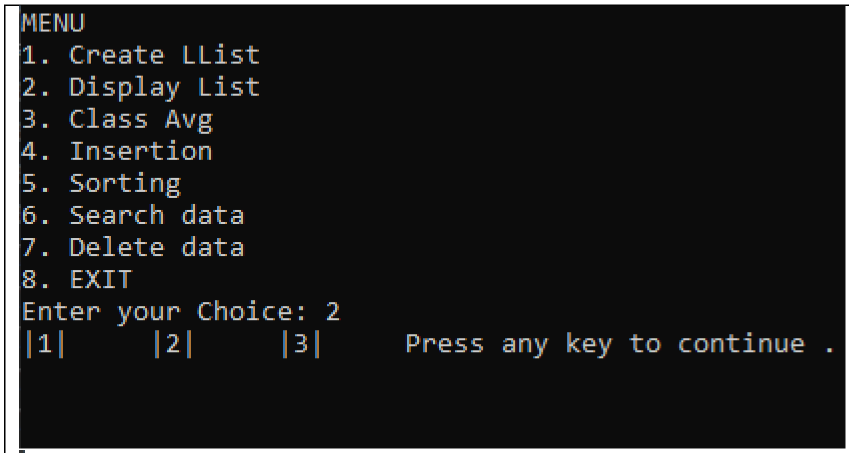
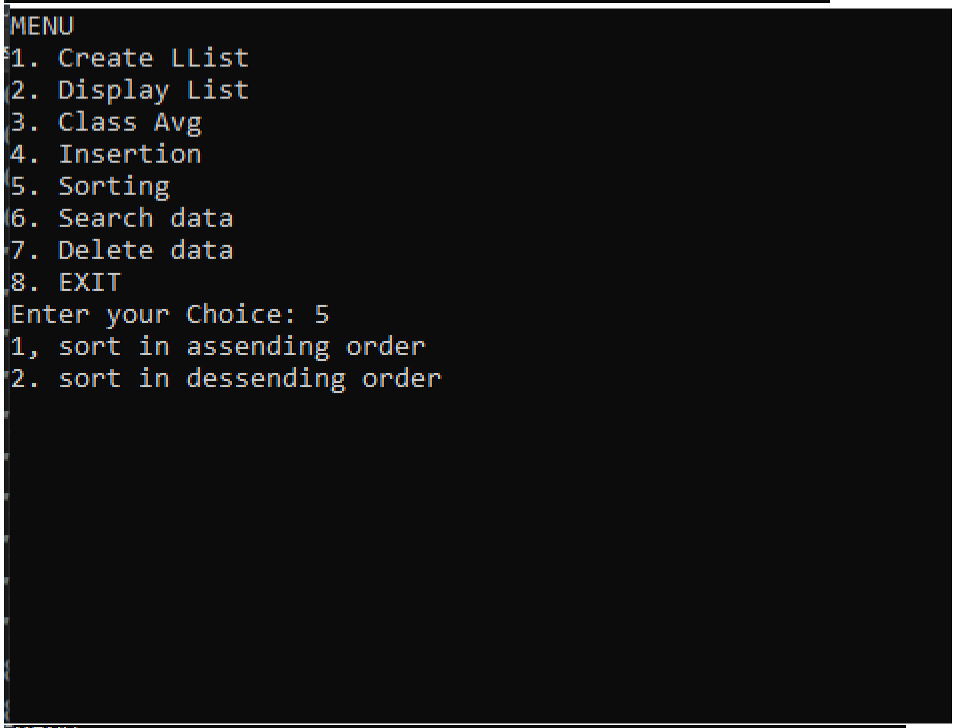
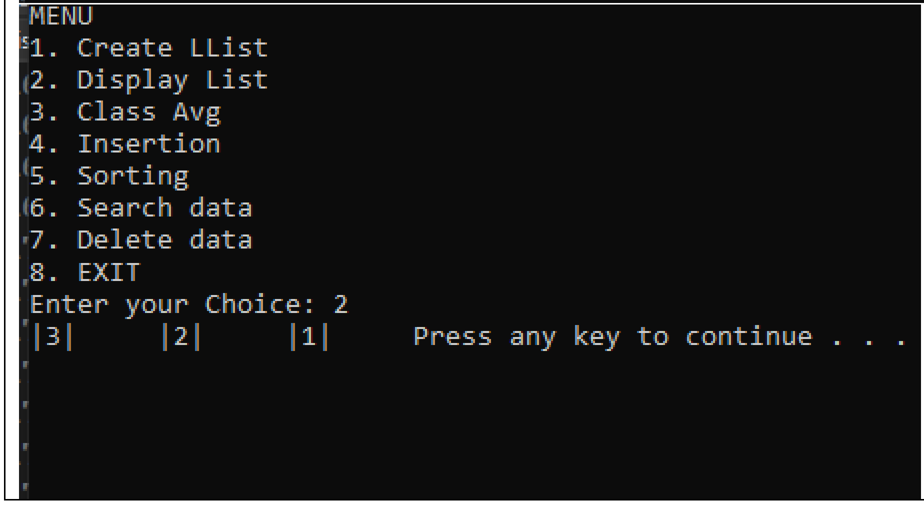
```
                        else
                                cout << "found at " << result;
                }
                break;
                case '7':
                        cout << "Which Node you want to delete? Enter Below" << endl;
                        cin >> no;
                        list.deleteLL(first, no);
                        break;

                case '8':
                        exit(-1);
                }
                system("pause");
                system("cls");
        } while (true);
        return 0;
}
```

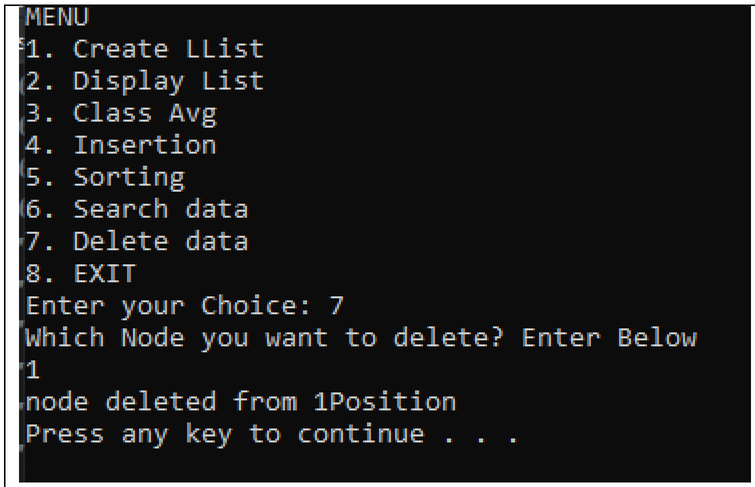


```
MENU
1. Create LList
2. Display List
3. Class Avg
4. Insertion
5. Sorting
6. Search data
7. Delete data
8. EXIT
Enter your Choice:
```



```
MENU
1. Create LList
2. Display List
3. Class Avg
4. Insertion
5. Sorting
6. Search data
7. Delete data
8. EXIT
Enter your Choice: 1
How many nodes you want to create
3
Enter Value
1
Enter Value
2
Enter Value
3
```


```
MENU
1. Create LList
2. Display List
3. Class Avg
4. Insertion
5. Sorting
6. Search data
7. Delete data
8. EXIT
Enter your Choice: 2
|1|     |2|     |3|     Press any key to continue .
```

```
MENU
1. Create LList
2. Display List
3. Class Avg
4. Insertion
5. Sorting
6. Search data
7. Delete data
8. EXIT
Enter your Choice: 5
1, sort in assending order
2. sort in dessending order
```

```
MENU
1. Create LList
2. Display List
3. Class Avg
4. Insertion
5. Sorting
6. Search data
7. Delete data
8. EXIT
Enter your Choice: 2
|3|     |2|     |1|     Press any key to continue . . .
```



```
MENU
1. Create LList
2. Display List
3. Class Avg
4. Insertion
5. Sorting
6. Search data
7. Delete data
8. EXIT
Enter your Choice: 7
Which Node you want to delete? Enter Below
1
node deleted from 1Position
Press any key to continue . . .
```