Assignment #10

Problem Solving and Programming in C++
Department of Computer Science
Old Dominion University

Objectives: After completing this assignment, students will able to:

- create a linked list – write the code to implement the list
- add elements to a linked list
- remove elements from a linked list
- search for elements in a linked list
- sort a linked list
- reverse a linked list
- use the standard library `<List>` and call many functions from this library
- compare between the created linked list and the standard library `<List>`

Background: Linked list is a way to store multiple objects of the same type. It is similar to array or vector in that it stores data in order but it accesses data via pointers unlike arrays. In a linked list, there is the pointer "head" that always points to first item in the list and the pointer "next" which points to the next item in the list. The pointer "current" is used to iterate over the linked list (or search the list). In order to iterate over the list, first, current is set to the value of head. When we are done with head, we write `current = current->next;` to move on to the next item in the list. We do this until `current->next == NULL;` which means we have reached the last item in the list. Keep in mind that this iteration doesn't apply to circular linked lists. When using pointers, we use "`->`" instead of "." To access the elements of an object.

General Instructions: Review the problem description below and implement this program in **C++**.
Your task is to create a Linked List making use of different functions, one method to create a linked list, one method to display the elements in the linked list, one method to implement the search functionality, one method to free the linked list.

Problem description: A University has a need to maintain student records such as their personal details and their grades. Currently they are maintaining these records on papers. But that involves a lot of manual work for their employees to store and fetch the records. Also sometimes they are facing some problems such as losing student records and taking a lot of time to search for a particular record. As the University is gaining more recognition, the number of students getting registered is increasing year by year. The University has a shortage of funds to hire more employees to maintain the records of student details.

The University Management decides to hire an application developer to design an application that will be able to store and search their students' records.

Application should be capable of storing student's personal records and their grades. Personal records include Student's First Name, Last Name, UIN, Date of Birth, and GPA.

Your task is to create the Double Linked Linked List which implements the following functions:

- Creates a Linked List to store student records.

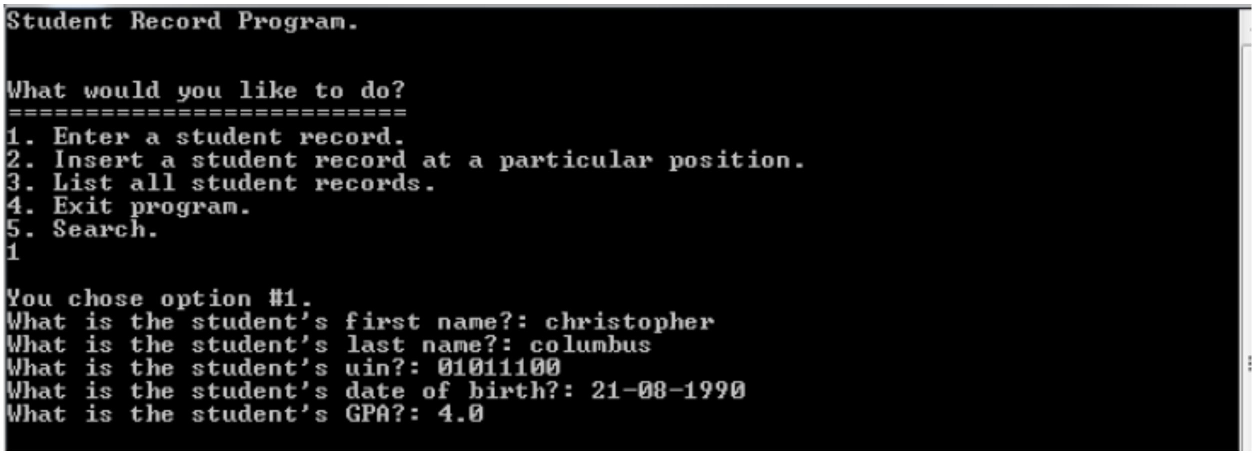
- Displays student records.
- Searches for a particular student record based on the Student UIN.
- Free the list on user request.

Resources Provided:

You are provided with a program that implements single linked list.
This program performs the following functions:

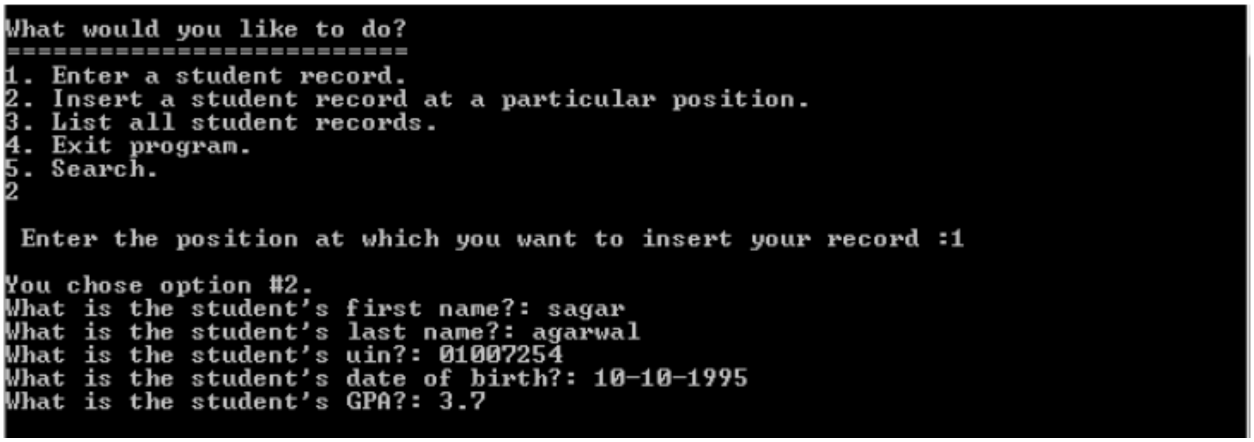
- Creates a Linked List to store student records.
- Displays the student records.
- Searches a particular student record based on the Student UIN.
- Free the list on user request.

Output



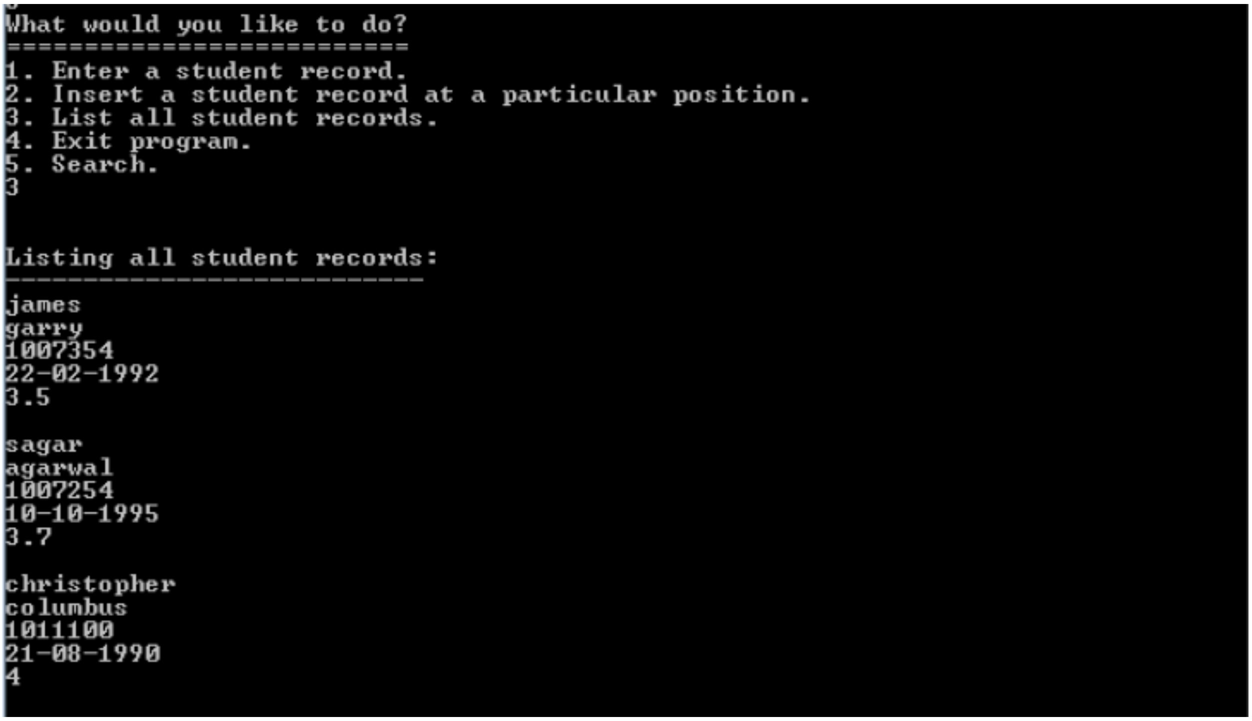
```
Student Record Program.


What would you like to do?
========================
1. Enter a student record.
2. Insert a student record at a particular position.
3. List all student records.
4. Exit program.
5. Search.
1

You chose option #1.
What is the student's first name?: christopher
What is the student's last name?: columbus
What is the student's uin?: 01011100
What is the student's date of birth?: 21-08-1990
What is the student's GPA?: 4.0
```

Fig 1. Console prompting user to enter student details after selecting option 1.



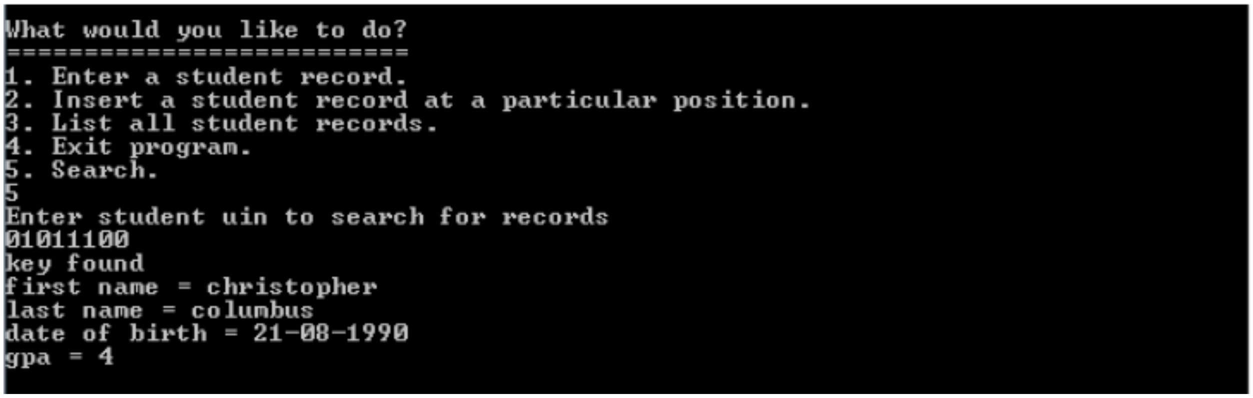
```
What would you like to do?
========================
1. Enter a student record.
2. Insert a student record at a particular position.
3. List all student records.
4. Exit program.
5. Search.
2

 Enter the position at which you want to insert your record :1

You chose option #2.
What is the student's first name?: sagar
What is the student's last name?: agarwal
What is the student's uin?: 01007254
What is the student's date of birth?: 10-10-1995
What is the student's GPA?: 3.7
```

Fig 2.Inserting student records at position 1.



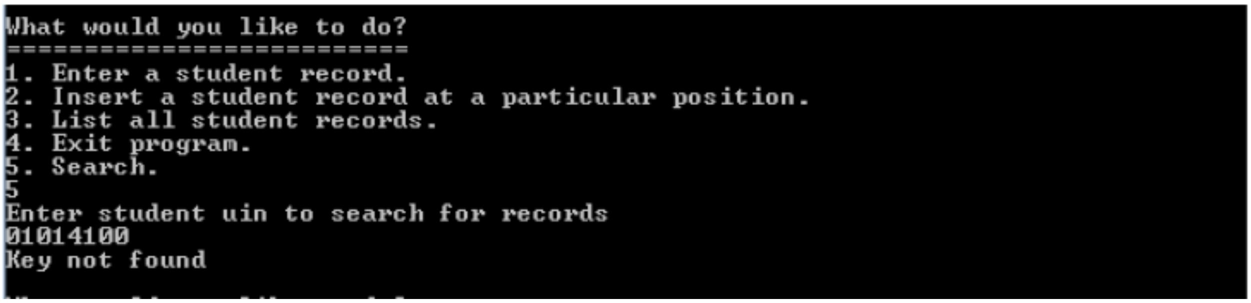
```
What would you like to do?
==========================
1. Enter a student record.
2. Insert a student record at a particular position.
3. List all student records.
4. Exit program.
5. Search.
3


Listing all student records:
---------------------------
james
garry
1007354
22-02-1992
3.5

sagar
agarwal
1007254
10-10-1995
3.7

christopher
columbus
1011100
21-08-1990
4
```

Fig 3. Displaying list of all students after selecting option 3.



```
What would you like to do?
==========================
1. Enter a student record.
2. Insert a student record at a particular position.
3. List all student records.
4. Exit program.
5. Search.
5
Enter student uin to search for records
01011100
key found
first name = christopher
last name = columbus
date of birth = 21-08-1990
gpa = 4
```

Fig 4. Searching for a student's record based on student's UIN (search successful)



```
What would you like to do?
==========================
1. Enter a student record.
2. Insert a student record at a particular position.
3. List all student records.
4. Exit program.
5. Search.
5
Enter student uin to search for records
01014100
Key not found
```

Fig 5. Searching for a student's record based on student's UIN (search unsuccessful)

Submission notes:

- Submit all files from your project including the .cpp and .h file(s).
- Zip all files and name it as "Assg10_cslogin-Task1", where the cslogin is your login ID for the account at the Department of Computer Science at ODU.

Task-2: Using the Standard Library – Linked List

Objectives:

Your task for this part of the lab is to use the standard library `<list>` to create 4 different linked lists:

1. Integer list
2. Float list
3. String list
4. A cellphone Contact list including name, telephone number, address, and further notes.

Background: In this portion of the lab, we need to use the standard library <list> which includes many helpful built-in functions. These functions perform different operations on lists, like sorting, searching, reversing, deleting, and adding elements to the list. An object of the class <list> is a container for storing a collection of objects (data items). Unlike the lists you create by hand, these lists come with pre-defined next and previous element pointers. By nature, lists are implemented as doubly linked lists. You can use an iterator to iterate through the list. Lists also support functions like push_back, push_front, insert, sort, and reverse. While the pointer pointing to the first item in the list can be accessed via `begin(),` the `end()` function returns a pointer to the spot after the last item in the list.

For more information on lists, what they are capable of, and how you can implement one, visit http://en.cppreference.com/w/cpp/container/list

Start:

1) Create the integer, and string lists as empty lists. Add 6 elements each to the lists based on:
   - The numerical order of the integer (between 1 and 10)
   - Alphabetical order of the string list

2) Create one more list of 1000 elements, all of them with the value 78.15.

3) Next, "push" 0 to the beginning and 15 to the end of the integer list.

4) Insert your name to the middle of the string list (at location 4) using the `insert()` functions

5) Add 2 to the first integer in the list and subtract 4 from the last integer in your integer list.

6) Iterate thorough your string linked list to print the values in the list.

7) Create and fill the cellphone-contact list with 500 names. Use the Random Number Generator to create First/Family names. For phone numbers, keep the area code 757 and again, use the random number generator to generate phone numbers for the contacts.

8) Lastly, sort the integer list and then reverse it

Now that you have used both versions of the linked list (Task-1 and Task-2), think about the advantages and disadvantages of each method. Which one would you prefer to use?

Submission notes:

- Submit all files from your project including the .cpp and .h file(s).
- Zip all files and name it as "Assg10_cslogin-Task2", where the cslogin is your login ID for the computers at the Department of Computer Science at ODU.
- Submit the .zip file in the respective Blackboard link.