Department of Information Systems and Technologies

CTIS 152 – Data Structures and Algorithms
Spring 2024 - 2025

Lab Guide #18 – Week 12 - 2

OBJECTIVE : Linked List Library, Dynamic Memory Allocation and exercises
Instructor : Serpil TIN **Assistants :** Berk ÖNDER, Hatice Zehra YILMAZ

Q1. You are supposed to create an integer **linkedList_int.h** file by writing the following functions to implement a linked list;

- **`getNode`** that allocates enough memory for an integer number node, and returns the address of it.
- **`addAfter`** that gets the address of a node and an integer number as parameters, and adds the new number after the specified node.
- **`addBeginning`** that gets the initial address of a list and an integer number as parameters, and adds the new number to the beginning of the list. The function will return the new initial address of the list.
- **`displayList`** that will display an integer list from the head to the last node on the screen. If the list is empty, display an appropriate message.

Write a C program that creates the linked list, reads positive integer numbers from the user, stores them in the linked list, and displays the content of the list.

Project Name: LG18_Q1
File Name: Q1.cpp

Example Run:

```
Enter a number: 24
Enter a number: 36
Enter a number: 74
Enter a number: 58
Enter a number: -52

24 -> 36 -> 74 -> 58 -> NULL
```

Q2. Write a C program that creates a **string** linked list by reading football player names from the user, finds the shortest name in the list, and inserts a given player name after the shortest name. The content of the linked list will be displayed before and after insertion.

Implement the necessary functions for a string linked list in the **linkedList_str.h** header file and in your source program.

Write the following functions;

- **`createList`** that reads several player names until “end” is given and inserts each word into the linked list. The function returns the initial address of the linked list.
- **`findShortest`** that finds and returns the node that consists of the shortest word.

Project Name: LG18_Q2
File Name: Q2.cpp

Example Run:

```
Enter a player name (end to stop): Fernandes
Enter a player name (end to stop): Icardi
Enter a player name (end to stop): Belozoglu
Enter a player name (end to stop): Dilmen
Enter a player name (end to stop): Ronaldo
Enter a player name (end to stop): end

Fernandes -> Icardi -> Belozoglu -> Dilmen -> Ronaldo -> NULL

Enter a name to insert after the shortest name: Messi

The shortest name in the list is: Icardi

Fernandes -> Icardi -> Messi -> Belozoglu -> Dilmen -> Ronaldo -> NULL
```

Q3. A coding championship is organized for the students from the vocational high school. The students' information is stored in the text file named **"students.txt"** with their **name**, **surname**, **country,** and **number of lines** information.

Implement the necessary functions for a structure linked list in the **linkedList_struct.h** header file and in your source program.

Write a C program that reads the information for several students from the text file into a linked list and displays the list of all students on the screen. Then, it finds and displays the winner of the championship. Also, it displays the information of the students from the specified country. If the searched country is NOT in the list, displays a warning message.

Write the following functions;

createList, displayList, searchCountry, findWinner

Project Name: LG18_Q3
File Name: Q3.cpp

Example Run #1:

```
List of ALL Students
Name     Surname    Country     #OfLines
******   ********   *********   ********
Martin   Gelbero    Ukrain        4387 ->
Leon     Reid       Ireland       1219 ->
Taegon   PArk       Korea         2990 ->
Walsh    Byrne      Australia     3182 ->
Aaron    Brown      Canada        1754 ->
Ameli    Margaret   England       2876 ->
Ramil    Guliyev    Turkey        4510 ->
Isla     Bethany    NewZeland      940 ->
Alonso   Edward     Panama        1348 ->
Noah     Lyles      Usa           3856 ->
NULL

The Winner: Ramil Guliyev, Turkey, 4510

Enter a country to search: England
Ameli Margaret, England, 2876
```

Example Run #2:

```
List of ALL Students
Name     Surname    Country     #OfLines
******   ********   *********   ********
Martin   Gelbero    Ukrain        4387 ->
Leon     Reid       Ireland       1219 ->
Taegon   PArk       Korea         2990 ->
Walsh    Byrne      Australia     3182 ->
Aaron    Brown      Canada        1754 ->
Ameli    Margaret   England       2876 ->
Ramil    Guliyev    Turkey        4510 ->
Isla     Bethany    NewZeland      940 ->
Alonso   Edward     Panama        1348 ->
Noah     Lyles      Usa           3856 ->
NULL

The Winner: Ramil Guliyev, Turkey, 4510

Enter a country to search: Cyprus
There is NO Student from Cyprus !
```

students.txt

```
Martin Gelbero Ukrain 4387
Leon Reid Ireland 1219
Taegon PArk Korea 2990
Walsh Byrne Australia 3182
Aaron Brown Canada 1754
Ameli Margaret England 2876
Ramil Guliyev Turkey 4510
Isla Bethany NewZeland 940
Alonso Edward Panama 1348
Noah Lyles Usa 3856
```

Additional Question

Write a C program that creates a **sorted** linked list, gets the integer numbers from the file, stores them in the linked list, and displays the content of the list. Do NOT forget to eliminate the **duplicate** values.

Implement the necessary functions for an integer linked list in the **linkedList_int.h** header file.

Write the following functions in your source program;

- **searchToInsert:** searches for the specified number, finds and returns the correct position in the linked list. If the searched number already exists in the list, the function returns NULL.
- **createList:** gets an input text file pointer as a parameter, reads the numbers from the text file, and inserts each number into the **correct** position in the linked list. The function returns the initial address of the linked list.
  Note: Write a search function to find the correct position in the linked list.

Project Name: LG18_AQ
File Name: AQ.cpp

numList.txt

```
248 25 174 334 373 10 174 28 124 257 152
```

Example Run:

```
List is: 10 -> 25 -> 28 -> 124 -> 152 -> 174 -> 248 -> 257 -> 334 -> 373 -> NULL
```