

# Department of Information Systems and Technologies

## CTIS 152 – Data Structures and Algorithms

FALL 2024 - 2025

### Lab Guide #19

**OBJECTIVE** : Linked List Operations

**Instructor** : Burcu LIMAN

**Assistant** : Berk ÖNDER, Engin Zafer KIRAÇBEDEL

**Q1.** You are supposed to create an integer **linked\_list.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. You can write this to the main part.

Write a C program that creates the linked list, reads several integer numbers from the user until -1 is entered, stores them in the linked list and displays the content of the list.

**Example Run:**

```
Enter a number: 9
Enter a number: 3
Enter a number: 7
Enter a number: 4
Enter a number: -1
9 -> 3 -> 7 -> 4 -> NULL
```

**Project Name:** LG19\_Q1

**File Name:** Q1.cpp

**Q2.** Write a C program that creates a **string** linked list by reading several words from the user, finds the longest word in the list and inserts a given word after the longest word. The content of the linked list will be displayed before and after insertion. Implement the necessary functions for a string linked list in the **linked\_list.h** header file and in your source program.

Write the following functions;

- **createList**, that reads several words until "end" is given and inserts each word to the linked list. The function returns the initial address of the linked list.
- **findLongest**, that finds and returns the address of the node which consists the longest word.

**Example Run:**

```
Enter a word: difficult
Enter a word: profit
Enter a word: feel
Enter a word: refurbished
Enter a word: puzzle
Enter a word: end
```

```
difficult -> profit -> feel -> refurbished -> puzzle -> NULL
Enter a word to insert after the longest word: phone
```

The longest word in the list is: refurbished

```
difficult -> profit -> feel -> refurbished -> phone -> puzzle -> NULL
```

**Project Name:** LG19\_Q2

**File Name:** Q2.cpp

**Q3.** An international athleticism championship is organized for 100 meter runners. The information of runners is stored in the text file named **runners.txt** with their **name, surname, age, country** and **score** information.

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

- Write the following functions:  
**createList, displayList, searchCountry, findGoldMedalWinner.**

**Example Run#1:**

```
List of ALL Runners
Name      Surname      Age      Country      Score
*****
Ramil     Guliyev      29      Turkey      1416 ->
Alonso    Edward       30      Panama      1348 ->
Noah      Lyles        22      Usa         1458 ->
Leon     Reid         25      Ireland     1219 ->
Taegon    PArk         28      Korea       1190 ->
Aaron     Brown        27      Canada      1354 ->
NULL
```

The Gold Medal Goes to: Noah Lyles, 22,Usa, 1458

Enter a country to search: Germany  
There is NO Runner from Germany !

**Runners.txt**

```
Ramil Guliyev 29 Turkey 1416
Alonso Edward 30 Panama 1348
Noah Lyles 22 Usa 1458
Leon Reid 25 Ireland 1219
Taegon PArk 28 Korea 1190
Aaron Brown 27 Canada 1354
```

**Example Run#2:**

```
List of ALL Runners
Name      Surname      Age      Country      Score
*****
Ramil     Guliyev      29      Turkey      1416 ->
Alonso    Edward       30      Panama      1348 ->
Noah      Lyles        22      Usa         1458 ->
Leon     Reid         25      Ireland     1219 ->
Taegon    PArk         28      Korea       1190 ->
Aaron     Brown        27      Canada      1354 ->
NULL
```

The Gold Medal Goes to: Noah Lyles, 22,Usa, 1458

Enter a country to search: Turkey

Ramil Guliyev, 29,Turkey, 1416

**Project Name:** LG19\_Q3  
**File Name:** Q3.cpp

## Additional Question

### AQ.

The purpose of the program is to obtain a **sorted linked list** consisting of words.

- Implement the necessary functions for a linked list in the **linkedList.h** header file.

Write a C program that reads the words from “**words.txt**”, create and display the content of the sorted linked list as in the example run. The program also reads the word to be deleted, from the user, deletes the nodes containing this word from the list. The updated list will be displayed at the end of the program.

Write the following functions;

- **createList**: gets an input text file pointer as parameter, read the words from the text file, and inserts each word to the correct position in the linked list. After each insertion, the content of the list will be displayed. The function returns the initial address of the linked list. Note: Write a search function to find the correct position in the linked list.
- **removeWord**: gets the initial address of the linked list and the word to be deleted as parameters, deletes all words which start with the given string in the linked list. The function also returns **1** or **0** depending on the success of the delete operation.

```
words.txt
acceptance
breeze
apple
backhand
accept
zoo
backward
bicycle
back
playback
```

#### Example Run #1:

```
acceptance -> NULL
acceptance -> breeze -> NULL
acceptance -> apple -> breeze -> NULL
acceptance -> apple -> backhand -> breeze -> NULL
accept -> acceptance -> apple -> backhand -> breeze -> NULL
accept -> acceptance -> apple -> backhand -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> backhand -> backward -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> backhand -> backward -> bicycle -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> back -> backhand -> backward -> bicycle -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> back -> backhand -> backward -> bicycle -> breeze -> playback ->
zoo -> NULL
```

Enter a string to delete: back

```
accept -> acceptance -> apple -> bicycle -> breeze -> playback -> zoo -> NULL
```

#### Example Run #2:

```
acceptance -> NULL
acceptance -> breeze -> NULL
acceptance -> apple -> breeze -> NULL
acceptance -> apple -> backhand -> breeze -> NULL
accept -> acceptance -> apple -> backhand -> breeze -> NULL
accept -> acceptance -> apple -> backhand -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> backhand -> backward -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> backhand -> backward -> bicycle -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> back -> backhand -> backward -> bicycle -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> back -> backhand -> backward -> bicycle -> breeze -> playback ->
zoo -> NULL
```

Enter a string to delete: accept

```
apple -> back -> backhand -> backward -> bicycle -> breeze -> playback -> zoo -> NULL
```

#### Example Run #3:

```
acceptance -> NULL
acceptance -> breeze -> NULL
acceptance -> apple -> breeze -> NULL
acceptance -> apple -> backhand -> breeze -> NULL
accept -> acceptance -> apple -> backhand -> breeze -> NULL
accept -> acceptance -> apple -> backhand -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> backhand -> backward -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> backhand -> backward -> bicycle -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> back -> backhand -> backward -> bicycle -> breeze -> zoo -> NULL
accept -> acceptance -> apple -> back -> backhand -> backward -> bicycle -> breeze -> playback ->
zoo -> NULL
```

Enter a string to delete: book

There is NO word which starts with the string <book>