Department of Information Systems and Technologies

**CTIS 152 – Data Structures and Algorithms**
Spring 2024 - 2025

**Lab Guide #20 – Week 14 - 1**

| **OBJECTIVE :** Hash tables with Linked Lists, Linked List Exercises |
|---|
| **Instructor :** Serpil TIN<br>**Assistants :** Berk ÖNDER& Hatice Zehra YILMAZ |

**Q1.** Computer center keeps the usernames and passwords in a hash table with size 10.
Write a C program that doing all of the inserting, removing, searching and displaying the users. It asks the username (usernames are unique) to the user and stores these information in a structure (username and password).

Write the following functions;

- **hashCode**: that finds the hash index by using the total of the ascii codes of each letter of the given username.
- **initArray**: takes the hash table and initializes it.
- **menu**: displays a menu, reads and validates the user's choice and then returns the choice.
- **searchUser** that takes Username and the address of the linked list as parameters, searches the username in the linked list and returns the address of the username. If the username does not exist in the list, it will return NULL.
- **insertUser** that takes user information and a hash table, inserts that info into the array using a linked list (addBeginning /addAfter). If the given username has already been exist in the linked list, it will update the password of already existing username. If username doesn't exist, it will add the username at the end of the linked list.
- **removeUser** that takes username and the hash table and removes that user from the hash table.
- **display** that takes the hash table as a parameter and displays the information of the users in the hash map.

**Project Name:** LG20_Q1
**File Name:** Q1.cpp

**Example Run:**

```
                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 1

Enter username and password : hyilmaz hb45aewq
Tot of ascii codes: 766
Inserted User : hyilmaz (username) and hb45aewq (password)

                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 3

Head [0] has no elements
Head [1] has no elements
Head [2] has no elements
Head [3] has no elements
Head [4] has no elements
Head [5] has no elements
Head [6] has elements : hyilmaz hb45aewq, NULL

Head [7] has no elements
Head [8] has no elements
Head [9] has no elements
                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 1

Enter username and password : yilmazh kuyt5aw8
Tot of ascii codes: 766
Inserted User : yilmazh (username) and kuyt5aw8 (password)
```

```
                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 1

Enter username and password : hyilmaz kuqw4567
Tot of ascii codes: 766

The username has been already inserted,
password will be updated!
Inserted User : hyilmaz (username) and kuqw4567 (password)

                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 1

Enter username and password : aliguler kiw4sg62
Tot of ascii codes: 853
Inserted User : aliguler (username) and kiw4sg62 (password)

                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 3

Head [0] has no elements
Head [1] has no elements
Head [2] has no elements
Head [3] has elements : aliguler kiw4sg62, NULL
Head [4] has no elements
Head [5] has no elements
Head [6] has elements : hyilmaz kuqw4567, yilmazh kuyt5aw8, NULL
Head [7] has no elements
Head [8] has no elements
Head [9] has no elements
                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 1

Enter username and password : cserim hy42wta8
Tot of ascii codes: 643
Inserted User : cserim (username) and hy42wta8 (password)
```

```
                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 3

Head [0] has no elements
Head [1] has no elements
Head [2] has no elements
Head [3] has elements : aliguler kiw4sg62, cserim hy42wta8, NULL
Head [4] has no elements
Head [5] has no elements
Head [6] has elements : hyilmaz kuqw4567, yilmazh kuyt5aw8, NULL
Head [7] has no elements
Head [8] has no elements
Head [9] has no elements
                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 2

Enter the username to delete : hyilmaz
Tot of ascii codes: 766

User which has the username hyilmaz is removed.

                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 3

Head [0] has no elements
Head [1] has no elements
Head [2] has no elements
Head [3] has elements : aliguler kiw4sg62, cserim hy42wta8, NULL
Head [4] has no elements
Head [5] has no elements
Head [6] has elements : yilmazh kuyt5aw8, NULL
Head [7] has no elements
Head [8] has no elements
Head [9] has no elements
                MENU
************************************
1. Insert a supply into the Hash Table
2. Remove a supply from the Hash Table
3. Display a Hash Table
4. Exit

Please enter your choice : 4
```

**Q2.** There are 3 types of courses in the gym. They are listed below according to the their ids;

1 -> Aerobics

2 -> Zumba

3 -> Pilates

Write a C program that reads a text file named **"courses.txt"** into an array of structure linked list. The program should create a linked list for every type of courses and fill these linked lists according to their ids. Then, it will displays the content of the linked list array which includes the courses' linked lists in the correct order.

**Note:** Initial address ( i.e, head pointer ) of each linked list must be stored in an array.

Write the following functions;

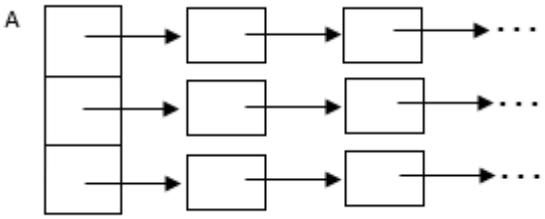
- **`createList:`** gets the file and a linked list array as parameters for reading the text file and fills the linked lists according to the courseIDs. The linked list will keep the courses in reverse order (each node will be added to the beginning of the linked list).
- **`displayList:`** gets a linked list as a parameter to display its' content.

**Courses.txt**

```
1 Abby Kohn : 3 months
3 Michael Pearce : 1 year
2 Dante Tomaselli : 1 month
2 Jeremy Dyson : 1 week
3 Alexandros Avranas : 3 months
1 Kay Cannon : 1 month
3 Timothy McNeil : 1 month
2 John Krasinski : 1 year
1 Holderman : 3 months
1 Raja Gosnell : 1 year
3 Dominic Cooke : 3 months
2 Leigh Whannell : 3 months
2 Ari Aster : 3 months
3 Paul Schrader : 1 year
1 Tim Kirby : 1 week
```

**Project Name:** LG20_Q2
**File Name:** Q2.cpp

**HINT:** `node_t * A[3];`




**Example Run :**

```
Aerobics
**************************************
   1  Tim Kirby            1 week     ->
   1  Raja Gosnell         1 year     ->
   1  Holderman            3 months   ->
   1  Kay Cannon           1 month    ->
   1  Abby Kohn            3 months   ->
NULL

Zumba
**************************************
   2  Ari Aster            3 months   ->
   2  Leigh Whannell       3 months   ->
   2  John Krasinski       1 year     ->
   2  Jeremy Dyson         1 week     ->
   2  Dante Tomaselli      1 month    ->
NULL

Pilates
**************************************
   3  Paul Schrader        1 year     ->
   3  Dominic Cooke        3 months   ->
   3  Timothy McNeil       1 month    ->
   3  Alexandros Avranas   3 months   ->
   3  Michael Pearce       1 year     ->
NULL
```

## Additional Question

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_str.h** header file.

Write the following functions;

- **`createList`** gets an input text file pointer as a parameter, reads 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.

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.

**Project Name:** LG20_AQ
**File Name:** AQ.cpp

**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>
```