

Department of Information Systems and Technologies

CTIS 152 – Data Structures and Algorithms

Fall 2024 - 2025

Lab Guide #12 – Week 9-1

OBJECTIVE : Bubble Sort, Binary Search

Instructor : Burcu LIMAN

Assistant : Berk ÖNDER - Engin Zafer KIRAÇBEDEL

Q1. Write a C program that reads the unsorted list of safety information of a company (gender, injured part of the body, incident type, incident cost) from the file **safety.txt** into a **dynamically** created structure array (the first line of the file consists the number of injuries) and sorts them according to the costs in descending order using bubble sort algorithm and displays the list on the screen.

Write the following functions; **readFromFile**, **display**, **bubbleSort**

Project_name: LG12_Q1

File_name: Q1.cpp

Example Run:

Gender	Injured PartOfBody	Injury type	Cost
female	abdomen	fallingObject	\$ 4831.00
male	hands	fallingObject	\$ 4618.00
male	multiple	equipment	\$ 2476.00
female	feet	lifting	\$ 470.00
male	arms	fall	\$ 430.00
female	legs	slip/trip	\$ 396.00
male	arms	burn	\$ 58.00

safety.txt

```
7
female legs slip/trip 396
male arms fall 430
male arms burn 58
female abdomen fallingObject 4831
male multiple equipment 2476
female feet lifting 470
male hands fallingObject 4618
```

BINARY SEARCH ALGORITHM:

1. Let top be the subscript of the initial array element.
2. Let bottom be the subscript of the last array element.
3. Repeat until top exceeds bottom, thus there are no more elements to check
4. Let middle be the subscript of the element halfway from top to bottom.
5. If the element at middle is the target, than return middle.
6. else if the element at middle is larger than the target, let bottom be middle-1, thus continue the search in the first half.
7. else let top be middle+1, thus continue the search in the second half.
8. Return -1 since the loop terminated, but the number is not found.

Q2. Write a C program that reads the course codes of a department from a sorted file **"courseCodes.txt"** into an integer array with the **maximum size 20**. Then, it gets a number from the user to search in the text file and displays the result as shown in the example run. Use binary search algorithm and display how many comparisons made to find the value. **(The Codes in the file are given in ascending order).**

Write the following functions;

- **readFromFile** that takes the file pointer, course code array as input parameters. Then, it reads the codes from the file into the array (MAX array size is 20) and returns the actual number of elements in the array.
- **binarySearch** that takes the course codes array, necessary index values of top and bottom parameters and a course code to search as parameters. Then searches the code in the array using the Binary Search Algorithm. The function should also return the number of search steps. **You may find the Binary search algorithm above ;)**

Example Run #1 :

Enter value to find : 365
365 is found at 9. line
2 comparisons.

Example Run #2 :

Enter value to find : 252
The number is NOT found in the courseCodes.txt
4 comparisons.

Example Run #3 :

Enter value to find : 222
222 is found at 5. line
4 comparisons.

courseCodes.txt

151
152
165
221
222
261
262
310
365
480
487
496

Project_name: LG12_Q2

File_name: Q2.cpp

Q3. Write a C program that reads the flower names from the “**flowers.txt**” file into a string array with the **maximum size 30**. Then the program reads a flower name to search from the user, and searches it through the flower list by using the **binary search** algorithm. (The list of the flowers in the file is given in ascending order). If the searched flower is found, an appropriate message will be displayed as in the example run. The program stops searching for flowers when the user enters “**end**” as a flower name.

Write the following functions: **readFromFile**, **display**, **binarySearch**

Project Name: LG12_Q3

File Name: Q3.cpp

Example Run:

The List of Flowers

African Lily
Begonia
Daffodil
Daisy
Evening Primrose
Forget-Me-Nots
Fresia
Gardenia
Hibiscus
Hydrangea
Iris
Jasmine
Lavender
Lilac
Lily
Magnolia
Orchid
Petunia
Rose
Sunflower
Tulip

Enter a flower name to search (or end to stop searching): African Lily
African Lily found on the index 0 in the list

Enter a flower name to search (or end to stop searching): Gardenia
Gardenia found on the index 7 in the list

Enter a flower name to search (or end to stop searching): Tulip
Tulip found on the index 20 in the list

Enter a flower name to search (or end to stop searching): Lily of Valley
Lily of Valley could not be found!

Enter a flower name to search (or end to stop searching): end

flowers.txt

African Lily
Begonia
Daffodil
Daisy
Evening Primrose
Forget-Me-Nots
Fresia
Gardenia
Hibiscus
Hydrangea
Iris
Jasmine
Lavender
Lilac
Lily
Magnolia
Orchid
Petunia
Rose
Sunflower
Tulip

AQ. Write a C program that forms a hotel list by getting hotel names from the text file “**hotels.txt**” into a string array with the **maximum size 30**, sorts the list in **ascending order** by using **bubble sort algorithm** and then displays the sorted list. The program displays the following menu and does the necessary operations according to the user’s choice;

```
MENU
*****
1. Display the Hotel List
2. Add a new Hotel
3. EXIT
```

When user selects “add a new hotel” option; the program asks for the new hotel name and checks the hotel if it is already in the list or not, displays a warning message or adds the new hotel into **correct position** in the list. The program stops when the user selects EXIT option from the menu.

Write the following functions;

- **menu**: displays a menu, read-validates and returns the user’s choice.
- **readFromFile**: reads the hotel names from the file into a string array and returns the actual size of the array.
- **display**: displays the hotel list.
- **bubbleSort**: sorts the hotel list in ascending order.
- **binarySearch**: searches for the specified hotel name in the hotel list and returns the index of the searched hotel. If the hotel is NOT in the list, functions returns -1 and the **correct position** to insert the hotel name.
- **shiftDown**: shifts down the hotels names to open a space to the given position.
- **addToList**: adds the new hotel to the hotel list using the function **shiftDown**.

Project Name: LG12_AQ
File Name: AQ.cpp

hotels.txt

```
Hilton Istanbul
Merit Crystal
JW Marriott Miami
Wow Cremlin Palace
Cambria Beach Lodge
Merit Royal
Hilton SAii Lagoon Maldives
```

Example Run:

```
Hotel List
-----
1) Cambria Beach Lodge
2) Hilton Istanbul
3) Hilton SAii Lagoon
  Maldives
4) JW Marriott Miami
5) Merit Crystal
6) Merit Royal
7) Wow Cremlin Palace

MENU
*****
1. Display the Hotel
  List
2. Add a new Hotel
3. EXIT
Enter your choice: 9

MENU
*****
1. Display the Hotel
  List
2. Add a new Hotel
3. EXIT
Enter your choice: -5
```

```
MENU
*****
1. Display the Hotel
  List
2. Add a new Hotel
3. EXIT
Enter your choice: 2

Enter the hotel name:
Hilton Istanbul
"Hilton Istanbul" already
exists in the list!

MENU
*****
1. Display the Hotel
  List
2. Add a new Hotel
3. EXIT
Enter your choice: 2

Enter the hotel name:
Disneyland Hotel
"Disneyland Hotel" ADDED
to the list!
```

```
MENU
*****
1. Display the Hotel
  List
2. Add a new Hotel
3. EXIT
Enter your choice: 1

Hotel List
-----
1) Cambria Beach Lodge
2) Disneyland Hotel
3) Hilton Istanbul
4) Hilton SAii Lagoon
  Maldives
5) JW Marriott Miami
6) Merit Crystal
7) Merit Royal
8) Wow Cremlin Palace

MENU
*****
1. Display the Hotel
  List
2. Add a new Hotel
3. EXIT
Enter your choice: 3
```

