**Faculty of Computers and Information – Fayoum University**

# Data Structures (AI & BIO) – Practical Project Sheet 2022

V1.1

| Course Title: | Course Team: |
|---|---|
| ACS242/BCS242: Data Structures<br>**AI & BIO Programs** | Dr. Basma Hassan (bhk00@fayoum.edu.eg)<br>Mahmoud Badry (mma18@fayoum.edu.eg)<br>Moataz Ahmed (maa60@fayoum.edu.eg) |
| **Project Title:** | **Project Submission Information** |
| Data Structures Playground | - This is a team-based project.<br>- Each Student must join a team of **6 members** at maximum.<br>- Each team should send **an email message** to **Mahmoud Badry** that includes the team members. Email must be sent before **22 May 2022**.<br>- Project will be discussed at Mahmoud or/and Moataz offices and will be uploaded to the Google classroom project's section. |
| **Submission Date:** | |
| **4 June 2022**<br>- Each team will have an announced **delivery time** that will be public on our Google's classroom | |
| **Marking and Assessment** | |
| This assignment will be marked out of 120%. 20% of the grade is an extra credit grade (Bonus) | |
| **Learning outcomes** | |
| - Try and Perfect Various data structures.<br>- Differentiate between Sequential data structures and Linked data structures.<br>- Learn how and where to use a specific data structure. | |

**IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED.**

**COPYING FROM EACH OTHER ISN'T ALLOWED.**

## I. Project – Introduction

There are various data structures that we have learned during the term. The project aims to show all these data structures in one application which will allow us to learn more about them and to know their differences.

The project should show a menu on a C++ console app for the user to try different data structures. In addition, there is an embedded phone directory application that acts like old mobiles on a console app.

**The better your application lives without crashing errors the better it will be.**

## II. Project – Menu structure and operations

The project menu will be as following (Shown in console):

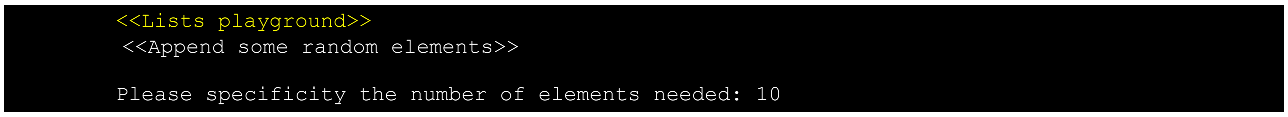
```
Welcome to the Data Structure playground project.
Choose an option from the following list:

A. Lists playground.
B. Stacks playground.
C. Queues playground.
D. Hashes playground.
E. Phone directory application
F. Show app calculations. (Bonus)
G. Exit application.

Please make a selection:
```

### A. Lists playground

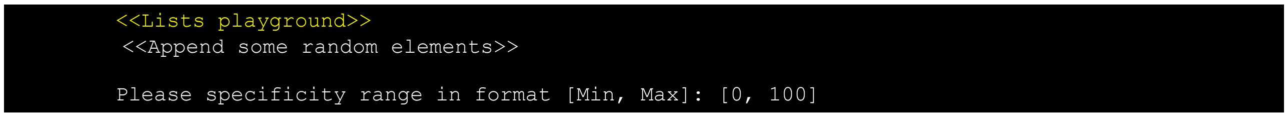
In the list's playground, three types of lists will be made in the background: Array-based list, a Linked list, and an ordered linked list. A timer must be made for any operation that a user can made and to be presented with the result for each list. **You will need to use the Header files that we have used during the term. They can be edited to fit the requirements if needed.**

The following scenario shows the requirements (Shown in console):

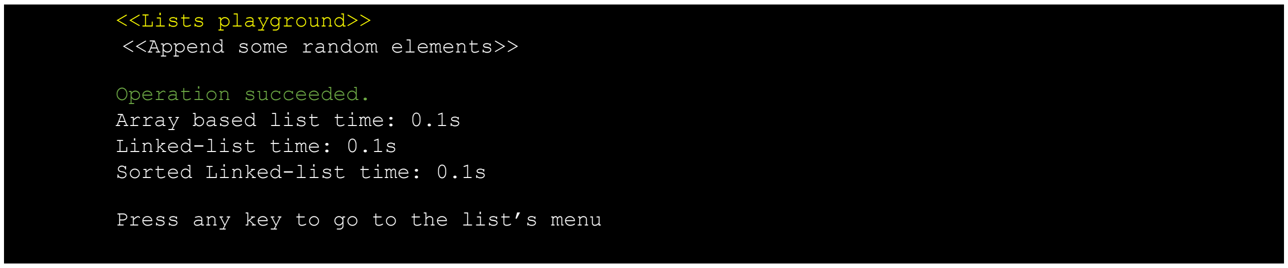
```
<<Lists playground>>

An integer Array based list, a linked list, and a sorted linked list have been created.

What are the operations that you need to do?

A. Append some random elements.
B. Append an element.
C. Insert an element at a specific place.
D. Delete an element from a specific place.
E. Delete a batch of elements.
F. Edit an element at a specific place.
G. Find an element.
H. Count the occurrence of an element.
I. Display lists
J. Sort by Merge Sort (Bonus)
1. Main Menu
0. Exit application.

Please make a selection: A
```



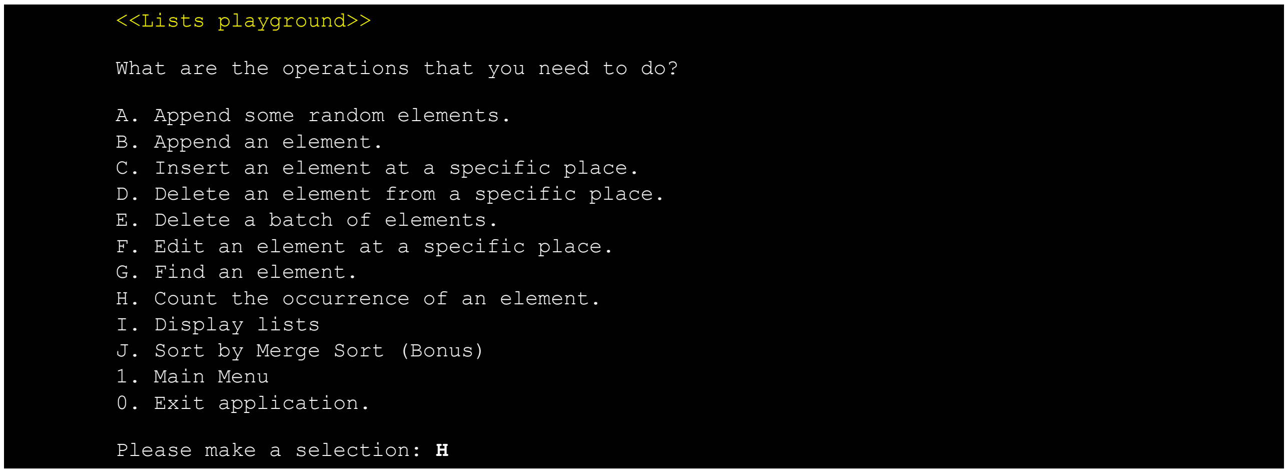
```
<<Lists playground>>
 <<Append some random elements>>

Please specificity the number of elements needed: 10
```



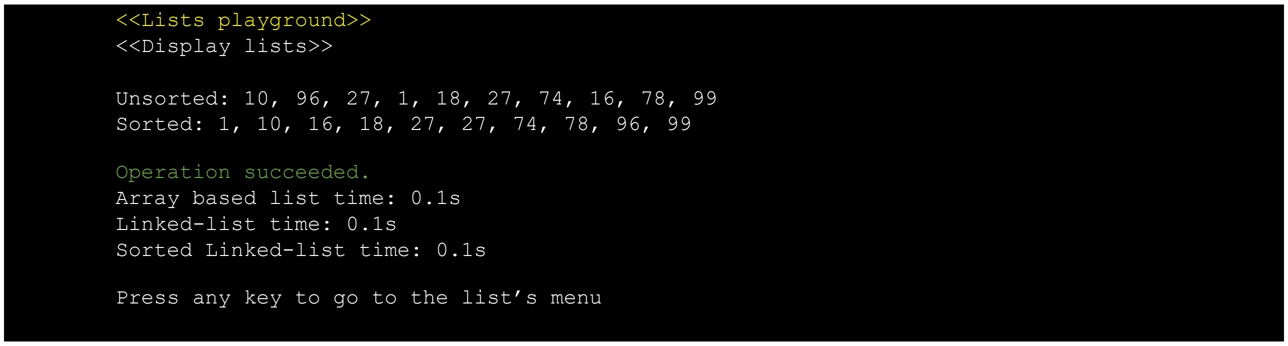
```
<<Lists playground>>
 <<Append some random elements>>

Please specificity range in format [Min, Max]: [0, 100]
```



```
<<Lists playground>>
 <<Append some random elements>>

Operation succeeded.
Array based list time: 0.1s
Linked-list time: 0.1s
Sorted Linked-list time: 0.1s

Press any key to go to the list’s menu
```



```
<<Lists playground>>

What are the operations that you need to do?

A. Append some random elements.
B. Append an element.
C. Insert an element at a specific place.
D. Delete an element from a specific place.
E. Delete a batch of elements.
F. Edit an element at a specific place.
G. Find an element.
H. Count the occurrence of an element.
I. Display lists
J. Sort by Merge Sort (Bonus)
1. Main Menu
0. Exit application.

Please make a selection: H
```



```
<<Lists playground>>
<<Display lists>>

Unsorted: 10, 96, 27, 1, 18, 27, 74, 16, 78, 99
Sorted: 1, 10, 16, 18, 27, 27, 74, 78, 96, 99

Operation succeeded.
Array based list time: 0.1s
Linked-list time: 0.1s
Sorted Linked-list time: 0.1s

Press any key to go to the list’s menu
```

## B. Stacks playground

In the stack's playground, two types of stacks will be made in the background: an Array-based stack and a linked-based stack. A timer must be made for any operation that a user can made and to be presented with the result for each list. **You will need to use the Header files that we have used during the term. They can be edited to fit the requirements if needed.**

The following scenario shows the requirements (Shown in console):

```
<<Stacks playground>>

An integer Array based stack and a linked based stack have been created.

What are the operations that you need to do?

A. Push some random elements.
B. Push an element.
C. Pop an element.
D. Pop some elements.
E. Clear stack.
F. Display stacks (Without effecting the created stacks)
G. Is Stack sorted (Ascending or Descending) (Bonus)
1. Main Menu
0. Exit application.

Please make a selection: B
```

```
<<stacks playground>>
<<Push an element>>

Please specificity the number you want to push: 10
```

```
<<stacks playground>>
<<Push an element>>

Operation succeeded.
Array based stack time: 0.1s
Linked-list stack time: 0.1s

Press any key to go to the stack's menu
```

## C. Queues playground

In the queue's playground, two types of queues will be made in the background: an Array-based queue and a linked-based queue. A timer must be made for any operation that a user can made and to be presented with the result for each list. **You will need to use the Header files that we have used during the term. They can be edited to fit the requirements if needed.**

The following scenario shows the requirements (Shown in console):

```
<<Queues playground>>

An integer Array based queue and a linked based queue have been created.

What are the operations that you need to do?
```

```
A. Add some random elements.
B. Add an element.
C. Delete an element.
D. Delete some elements.
E. Clear queue.
F. Display queue (Without effecting the created stacks)
G. Move Nth Element to be first (Bonus)
1. Main Menu
0. Exit application.

Please make a selection: A
```

```
<<Queues playground>>
 <<Add some random elements>>

Please specificity the number of elements needed: 5
```

```
<<Queues playground>>
 <<Add some random elements>>

Please specificity range in format [Min, Max]: [0, 100]
```

```
<<Queues playground>>
<<Add some random elements>>

Operation succeeded.
Array based queue time: 0.1s
Linked-list queue time: 0.1s

Press any key to go to the queue’s menu
```

```
<<Queues playground>>

What are the operations that you need to do?

A. Add some random elements.
B. Add an element.
C. Delete an element.
D. Delete some elements.
E. Clear queue.
F. Display queue (Without effecting the created queues)
G. Move Nth Element to be first (Bonus)
1. Main Menu
0. Exit application.

Please make a selection: F
```

```
<<Queues playground>>
<<Display queue>>

Unsorted: 10, 96, 27, 1, 18


Operation succeeded.
```

```
Array based queue time: 0.1s
Linked-based queue: 0.1s

Press any key to go to the queue’s menu
```

### Notes

- Move Nth Element to be first should move the element on N to be the first in the queue. For example, if the queue contains 1,2,3,4,5 and N=3. Then, the queue will be 3,1,2,4,5

## D. Hashes playground

In the hashes’ playground, one object of standard library’s “Map” will be created. A timer must be made for any operation that a user can made and to be presented with the result for each list.

**The hash key will be the same as the value inserted.**

```
<<Hashes playground>>

A map with an integer keys and values has been created.

What are the operations that you need to do?

A. Add some random elements.
B. Add an element.
C. Delete an element.
D. Delete the first N elements.
E. Find an element.
F. Display hash
G. Use the current hash to get unique numbers from the previous list. (Bonus)
1. Main Menu
0. Exit application.

Please make a selection: A
```

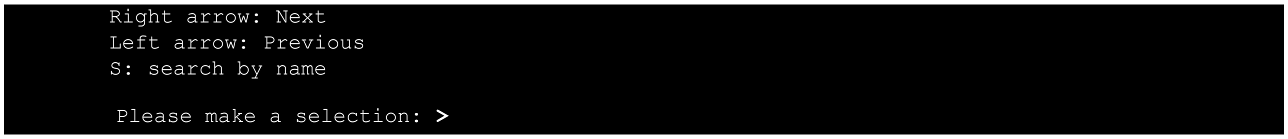
### Notes

- “`Use the current hash to get unique numbers from the previous list`” should work only if the lists background was used by a user and that it has some items. This operation should clear the current hash and get only unique values form the previously created list.

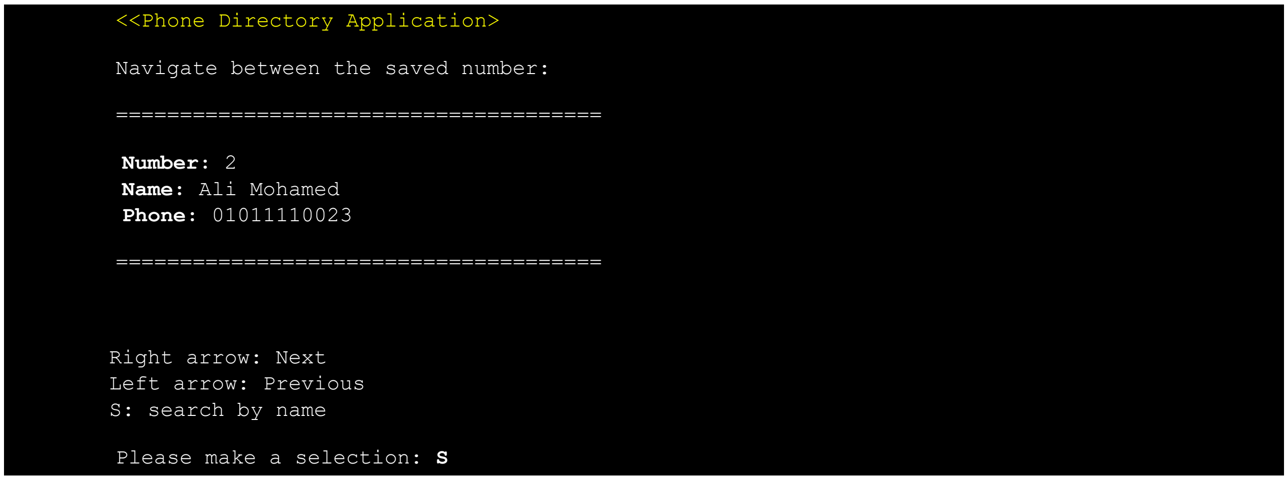
## E. Phone directory application

The phone directory application that acts like old mobiles on a console app. It will contain some **static** data that can be navigated between and a search function. The phone directory has to use **Doubly linked list.**

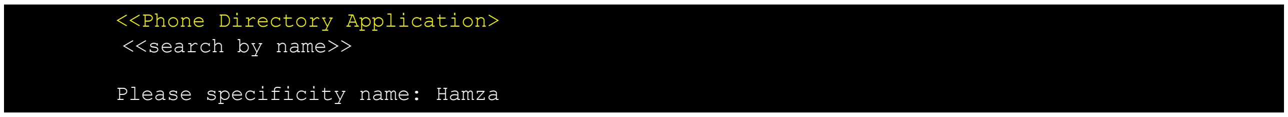
```
<<Phone Directory Application>

Navigate between the saved number:

======================================

 Number: 1
 Name: Ahmed
 Phone: 01011110022

======================================
```



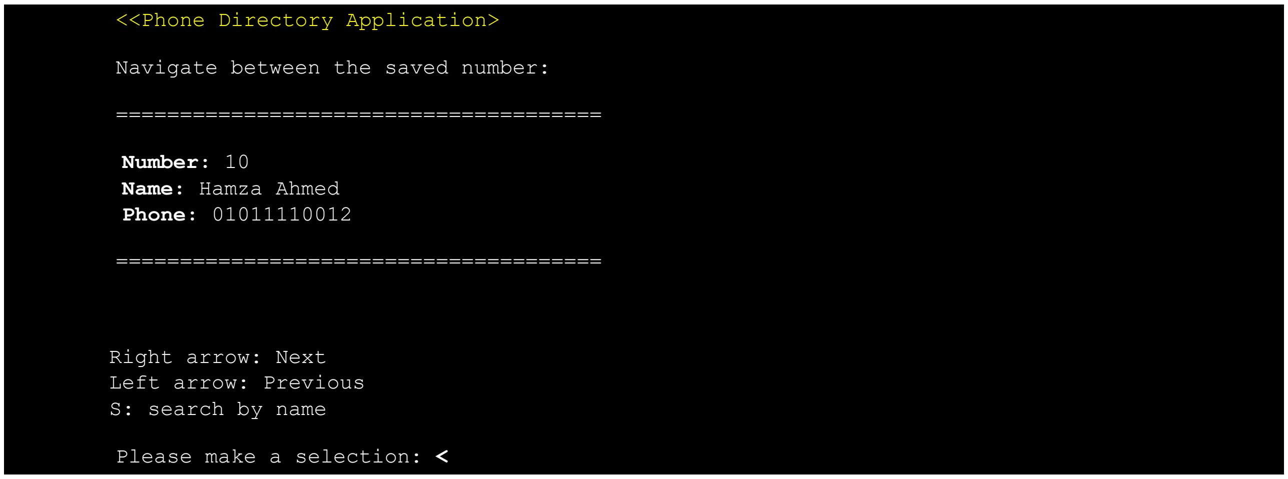
```
Right arrow: Next
Left arrow: Previous
S: search by name

 Please make a selection: >
```



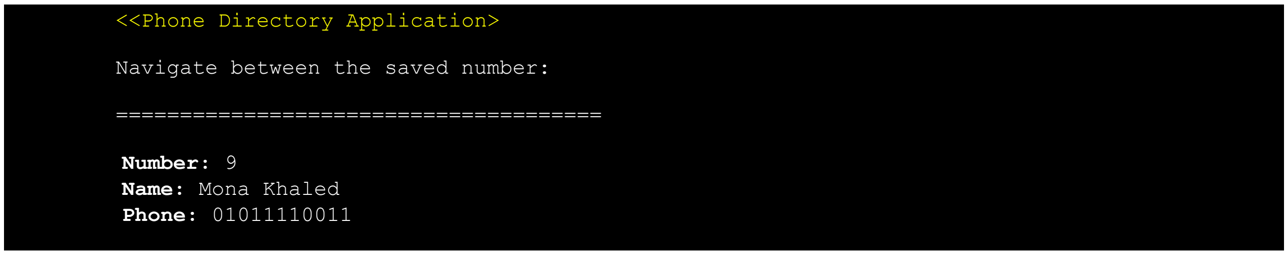
```
 <<Phone Directory Application>

 Navigate between the saved number:

 ======================================

  Number: 2
  Name: Ali Mohamed
  Phone: 01011110023

 ======================================



Right arrow: Next
Left arrow: Previous
S: search by name

 Please make a selection: S
```



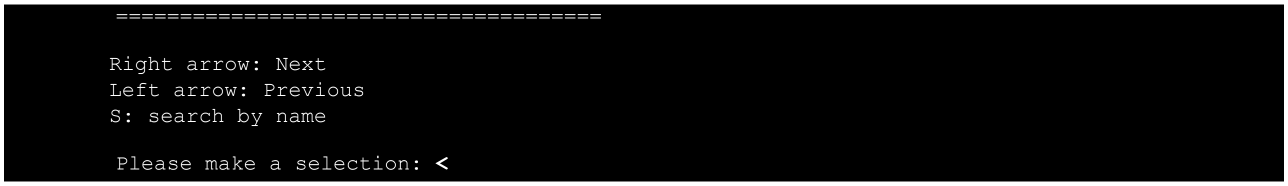
```
 <<Phone Directory Application>
  <<search by name>>

 Please specificity name: Hamza
```



```
 <<Phone Directory Application>

 Navigate between the saved number:

 ======================================

  Number: 10
  Name: Hamza Ahmed
  Phone: 01011110012

 ======================================



Right arrow: Next
Left arrow: Previous
S: search by name

 Please make a selection: <
```



```
 <<Phone Directory Application>

 Navigate between the saved number:

 ======================================

  Number: 9
  Name: Mona Khaled
  Phone: 01011110011
```



```
 =====================================

Right arrow: Next
Left arrow: Previous
S: search by name

 Please make a selection: <
```

### F. Show app calculations (Bonus)

This section of the program shows all the operations that have been created by the user during a specific application session. Each operation will be followed by the time needed for the operation to be made.

The data needed to be shown as following:

| **ID** | **Time** | **Scope** | **Data structure** | **Operation** | **Time taken** |
|---|---|---|---|---|---|
| 1 | 11:00AM | Lists | Array based list | Append some random values | 0.1S |
| 2 | 11:00AM | Lists | Linked list | Append some random values | 0.1S |
| 3 | 11:00AM | Lists | Sorted linked list | Append some random values | 0.1S |
| 4 | 11:10AM | Hashes | MAP | Append a value | 0.1S |
| 5 | 11:15AM | Stacks | Array based stack | Push an element | 0.1S |
| 6 | 11:15AM | Stacks | Linked-based stack | Push an element | 0.1S |

## III. Project – Technologies/Libraries

- C++ console application