Laboratory 2

Ordered List ADT

I. OBJECTIVES:

In this laboratory, you will

- apply the concepts of Ordered List ADT
- implement an array-based Ordered List ADT
- and create an ADT Ordered List.

II. DISCUSSION:

A useful tool for specifying the logical properties of a data type is the abstract data type, or ADT. A data type is a collection of values and a set of operations on those values. The term "abstract data type" refers to the basic mathematical concept that defines the data type. The ADT is a useful guideline to implementors and a useful tool to programmers who wish to use the data type correctly by specifying the mathematical and logical properties of a data type or structure.

Ordered List ADT

A sequence of elements together with these operations:

- Initialize the list.
- Creates an empty ordered list OL.
- Count the number of elements in the list.
- Insert an item anywhere in the list.
- Delete an item anywhere in a list.

Grocery List ADT

Grocery List ADT is one of the applications of Ordered List ADT. It is described as a collection of grocery items that can perform the following operations: count the items, insert items, delete items and display the items on the list.

Ordered List ADT Specifications

Value Definition:

abstract typedef <integer, array of itemType> ORDLIST;

Operator Definition:

abstract **CreateOrderedList**(OL) //*Creates an empty ordered list OL.*
Postcondition: OL.Size = 0

abstract **OrderedListLength**(OL)
//*Returns the number of items that are in the ordered list OL.*

Postcondition: OrderedListLength = OL.Size

abstract **OrderedListInsert**(OL, Position, NewItem)
//Inserts NewItem at position Position of ordered list OL

Precondition: The list is not full
Postcondition:
NewItem => Position
PrevItem[position] => Position + 1
PrevItem[position+1] => Position + 2
: :
PrevItem[position +(n-1)] => Position + n

abstract **OrderedListDelete**(OL, Position)
//Inserts NewItem at position Position of ordered list OL

Precondition: The list is not empty
Postcondition:
PrevItem[position] => deleted
PrevItem[position+1] => Position
PrevItem[position+2] => Position + 1
: :
PrevItem[position +n] => Position + (n-1)

Notice that the specifications of the operations contain no mention of how to store the ordered list or how to perform the operations.

III. TEST & DEBUG:

Step 1: Open the C program file **lab2sp1.c** located at *c:\datsala\lab2*.
Step 2: Compile and execute the program that will implement the following operator definitions using array:
abstract **CreateOrderedList**(OL)
abstract **OrderedListLength**(OL)
abstract **OrderedListInsert**(OL, Position, NewItem)
abstract **OrderedListDelete**(OL, Position)
abstract **OrderedListDisplay**(OL)

Note: *The library of LIST functions is* located at *c:\datsala\lab2 (filename: list.c)*

Step 3: Test each operation by invoking the following commands:

Command	Operations / Actions
- e	creates an empty (E) ordered list
- l	returns the length(L) of the list
- i Pos Item	inserts(I) the item at the given position (Pos)
- d Pos	delete (d) the items located at the given position (Pos)
- s	display on the screen (s) the items on the list

Do the following tests (Test Runs) and observe the output:

No	Command
1.	- e
2.	- i 1 milk
3.	- l
4.	- s
5.	- i 1 bread
6.	-s
7.	- i 2 eggs
8.	- s
9.	- d 2
10.	- s
11.	- d 5
12.	- l
13.	- e
14.	- d 1
15.	insert 11 items using *i*

Step 4: Construct the given grocery list using the ordered list operations

L = (milk, eggs, butter, apples, bread, chicken)

Step 5: Insert the item "nuts" after item butter using the ordered list operations.

Step 6: Delete the item "chicken" using the ordered list operations.

Step 7: Display the grocery list.

Step 8: Evaluate your test runs and provide analysis of results.

IV. SUPPLEMENTARY EXERCISES:

1. Create an ADT for the following operations:
   a. **DeleteAll**(*OL*) - Delete all items in the list.
   b. **InsertTwo**(*OL,item1,item2*) - Insert 2 items at the beginning of the list.

2. Implement the ADTs created in item IV.1 using the following commands:

Command	Operations / Actions
- d A	Delete all items in the list OL
- i 2 item1 item2	Insert items *item1* and *item2* in the list OL

V. MACHINE PROBLEM:

PROJECT: An Appointment Book

DESCRIPTION: A sequence of appointments that can perform the following operations: count the number of appointments for each month, insert an appointment, delete an appointment and display the appointments for each month on the list.

OPERATIONS:

1. Create an empty appointment list.
2. Returns the total appointments on the list for each month.
3. Inserts a new appointment at the given position.
4. Deletes an appointment located at the given position.
5. Display the appointments (including the month and day) on the list for each month.

REQUIREMENTS:

1. ADT Specifications.

   Consider this structure when designing your ADT.

   ```
   Month       : 1
   Day         : 21
   Appointment : Department's Meeting
   ```

2. Array-based implementation of ADT.
3. Test runs of the operations. Provide commands to invoke the operations.

Note: *The instructor may change the set of machine problems.*

Ordered List ADT

DATA & RESULTS SHEET

(Tentative Laboratory Report)

Name: _____

Schedule:_____ Section:_____

Date: _____ Grade: _____

Test & Debug:

<table>
<tr><th>Step No.</th><th>Answers/Results</th></tr>
<tr><td>3</td><td>
<table>
<tr><th>No</th><th>Command</th><th>Output</th></tr>
<tr><td>1.
2.
3.
4.
5.
6.
7.
8.
9.
10.
11.
12.
13.
14.
15.</td><td>- e
- i 1 milk
- l
- s
- i 1 bread
-s
- i 2 eggs
- s
- d 2
- s
- d 5
- l
- e
- d 1
insert 11 items using i</td><td></td></tr>
</table>
</td></tr>
<tr><td>4</td><td></td></tr>
<tr><td>5</td><td></td></tr>
</table>

6	
7	
8	

Note: For steps 4 to 7 provide the updates of the grocery list. Example: L (milk, bread)

Supplementary Exercises:

Supplementary Problem No.	Completed?		Remarks
	YES	NO	
1.a			
1.b			
2			

Machine Problem:

Requirements	Completed?		Remarks
	YES	NO	
1			
2			
3			

Note: Check the column YES if the requirement is provided otherwise check the column NO.(for instructors only)

INSTRUCTOR'S SIGNATURE: _____