Laboratory 5

Linked List

I. OBJECTIVES:

In this laboratory, you will

- understand and apply the concepts of the different types of linked list; singly linked list, doubly linked list and circular linked list
- implement Stack and Queue ADT using singly linked list.
- create a program that will simulate Josephus Problem.

II. DISCUSSION:

A **linked structure** is a collection of nodes storing data and links to other nodes (NODE PTR). The limitations of array-based implementation are; (1) its size has to be known at compilation time and (2) the data in the array are separated in computer memory by the same distance, which means that inserting an item inside the array requires shifting other data in this array. There are 3 types of linked list, namely (1) Singly Linked List, (2) Doubly Linked List and (3) Circular Linked List.

Singly Linked List Operations

(1) Insert(SL, pos, X)

Adds item X into the list(SL) at the given position (pos).

(2) Remove(SL,pos)

Deletes a node from a given list (SL) pointed to by the given position pos.

(3) Empty(SL)

Return TRUE if the list is empty, otherwise will return a value FALSE.

Doubly Linked List Operations

(1) Insertright(p, X)

Inserts a node with information field **X** to the right of the node pointed to by **p** in a doubly linked list.

(2) Delete(p)

Deletes the node pointed to by *p* from a doubly linked list and stores its contents in *x*. *Example: x=Delete(p)*

(3) Insertleft(p, X)

Inserts a node with information field **X** to the left of the node pointed to by **p** in a doubly linked list.

Circular Linked List Implementation

Josephus Problem:

This problem presents a group pf soldiers surrounded by an overwhelming enemy force. There is no hope of victory without reinforcements, but there is only a single horse available for escape. The soldiers agree to a pact to determine which of them is to escape and summon help. They form a circle and a number of **n** is picked from a hat. One of their names is also picked from a hat. Beginning with the soldier whose name is picked, they begin to count clockwise around the circle. When the count reaches, **n**, the soldier is removed from the circle, and the count begins again with the next soldier. The process continues so that each time the count reaches **n**, another soldier is removed from the circle. Any soldier removed from the circle is no longer counted. The last soldier remaining is to take the horse and escape. The problem is, given a number n, the ordering of the soldiers in the circle, and the soldiers from whom the count begins, to determine in which soldiers are eliminated from the circle and which soldier escapes.

The solution to this problem can be solved in a straight forward manner using circular linked list.

III. Test and Debug

1. Create a test program that will implement the following Linked List operations:

Type of Linked List	Operations

Singly Linked List	`Insert(SL, pos, X)`
	`Remove(SL,pos)`
	`Empty(SL)`
	`*Initialize (SL)`

***Initialize** operation makes the linked list an empty list.*

***Note**:The library of Linked List functions is available in c:\datsala\lab5*

2. Compile and test each operation using the following commands:

SINGLY LINKED LIST OPERATIONS:

Commands	Operation
> e	Empty(SL)
> i	Initialize (SL)
> + pos X	Insert(SL, pos, X)
> - pos	Remove(SL, pos)
> q	Exit the test program

Do the following sample runs:

SINGLY LINKED LIST OPERATIONS:

Commands	Results/Outputs
> i	Empty List
> - 0	Underflow!
> + 0 D	D
> + 1 L	DL
> + 2 S	DLS
> + 3 U	DLSU
> e	FALSE
> - 1	DSU
> - 2	DS
> - 0	S
> q	Exit program

* Display the prompt ">"

IV. SUPPLEMENTARY EXERCISES:

Consider a doubly linked list in which each node holds the pointer information, a pointer to the predecessor(left) and a pointer to the successor(right).

1. Modify the singly linked list program created in test & debug to simulate a doubly linked list operations

DOUBLY LINKED LIST OPERATIONS:

Commands	Operation
> i	`Initialize(p)`
> + L X	`Insertleft(p, X)`
> + R X	`Insertright(p, X)`
> -	`Delete(p)`
> q	`Exit the test program`

***Note**:The library of Linked List functions is available in c:\datsala\lab5*

2. and do the following test runs:

DOUBLY LINKED LIST OPERATIONS:

Commands	Results/Outputs
> i	Empty List
> -	Underflow!
> + R U	U
> + L S	SU
> + L L	LSU
> + L D	DLSU
> e	FALSE
> -	LSU
> -	SU
> -	U
> -	Empty
> q	Exit program

*Display the prompt ">"

V. Machine Problem

1. Write a program that will simulate the following using Linked list:

A. Conversions from infix to postfix notations (without parenthesis)

Given Infix expression: **A + B * C**
Simulation:

	SYMB	POSTFIX STRING	OPSTK
1	A	A	
2	+	A	+
3	B	AB	+
4	*	AB	+*
5	C	ABC	+*
6		ABC*	+
7		ABC*+	

B. Queue operations

Commands	Output
> E	True
> + A	A
> + B	A B
> + C	A B C
> -	B C
> + D	B C D
> + E	B C D E
- - (perform 5x)	Underflow

C. Josephus Problem
Given: n = 3 , name = A , Soldiers = ABCDE
Simulation:

Head	Circular Link List	Output
A	ABCDE	C
D	ABDE	A
B	BDE	E
B	BD	B
D	D	D

Note: *The instructor may change the machine problem.*

Linked List

DATA & RESULTS SHEET

(Tentative Laboratory Report)

Name: _____

Schedule:_____ Section:_____

Date: _____ Grade: _____

Test & Debug:

Step No.	Answers/Results
1	Attach source codes
2	SINGLY LINKED LIST OPERATIONS:

Commands	Results/Outputs
> i	Empty List
> - 0	Underflow!
> + 0 D	D
> + 1 L	DL
> + 2 S	DLS
> + 3 U	DLSU
> e	FALSE
> - 1	DSU
> - 2	DS
> - 0	S
> q	Exit program

Supplementary Exercises:

<table>
<tr><th>Supp No.</th><th>Answers/Results</th></tr>
<tr><td>1</td><td>Attach source codes</td></tr>
<tr><td>2</td><td>DOUBLY LINKED LIST OPERATIONS:
<table>
<tr><th>Commands</th><th>Results/Outputs</th></tr>
<tr><td>> i</td><td>Empty List</td></tr>
<tr><td>> -</td><td>Underflow!</td></tr>
<tr><td>> + R U</td><td>U</td></tr>
<tr><td>> + L S</td><td>SU</td></tr>
<tr><td>> + L L</td><td>LSU</td></tr>
<tr><td>> + L D</td><td>DLSU</td></tr>
<tr><td>> e</td><td>FALSE</td></tr>
<tr><td>> -</td><td>LSU</td></tr>
<tr><td>> -</td><td>SU</td></tr>
<tr><td>> -</td><td>U</td></tr>
<tr><td>> -</td><td>Empty</td></tr>
<tr><td>> q</td><td>Exit program</td></tr>
</table></td></tr>
</table>

Machine Problem:

Topic	Completed?		Remarks
	YES	NO	

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

A. Stack Application

	SYMB	POSTFIX STRING	OPSTK
1	A	A	
2	+	A	+
3	B	AB	+
4	*	AB	+*
5	C	ABC	+*
6		ABC*	+
7		ABC*+	

B. Queue Operations

Commands	Output
> E	True
> + A	A
> + B	A B
> + C	A B C
> -	B C
> + D	B C D
> + E	B C D E
- - (perform 5x)	Underflow

C. Josephus Problem

Given: n = 3 , name = A , Soldiers = ABCDE
Simulation:

Head	Circular Link List	Output
A	ABCDE	C
D	ABDE	A
B	BDE	E
B	BD	B
D	D	**D**

INSTRUCTOR'S SIGNATURE: _____