EAST WEST UNIVERSITY
Department of Computer Science and Engineering
B.Sc. in Computer Science and Engineering Program
CSE207: Data Structures

Assignment : **01**
Instructor : **Ahmed Abdal Shafi Rasel, Lecturer, Department of CSE**
Section : **06**
Trimester : **Spring 2025**

Objective: The objective of this assignment is to help students understand the implementation of the List Abstract Data Type (ADT) using both Array and Linked List representations.

Task: Create a List ADT with the following operations.

Operation	Description
1. void add(int item)	add item to the end of the List
2. void addbegin(int item)	Add item at the beginning of the list
3. void addanypos(int pos, int item)	Add item at position pos in the List, moving the items originally in positions pos .
4. bool contains(int item)	return true iff item is in the List
5. int size()	return the number of items in the List
6. bool isEmpty()	return true iff the List is empty
7. int get(int pos)	return the item at position pos in the List (error if pos is less than 0 or greater than or equal to size())
8. int indexOf(int item)	Return position of the specified element in the list
9. void removefirst()	Remove first element of the list
10. void removelast()	Remove last element of the list
11. void remove(int pos)	remove and return the item at position pos in the List
12. void reverse()	Reverse the element of the list
13. void sort()	Sort the element of the list in ascending order

Implementation Requirements:

1. Array-based List Implementation:
   - Use a dynamic array to store elements.
   - Handle resizing (dynamic allocation and deallocation) as elements are added or removed.
   - Ensure efficient implementation of operations such as insertion and deletion.
2. Linked List-based Implementation:
   - Implement a singly linked list with nodes storing data and a pointer to the next node.

- Implement efficient operations for inserting, deleting, and traversing elements.
- Compare the performance of Linked List vs. Array-based implementations.

Students must submit the **code implementation** along with **a short report** (1-2 pages) comparing the two implementations, discussing their complexity, advantages, and disadvantages, and highlighting key differences in performance and memory usage.