

### Sheet #4 (Linked Lists)

1. Implement the List ADT using an array (not linked).
2. Implement StackADT as a linked list.
3. Re-solve sheet 2 but for LinkedStackADT.
4. Implement QueueADT as a linked list.
5. Re-solve sheet 3 but for LinkedStackADT.
6. Write the function void JoinList (List \*pl1, List \*pl2) that copies all entries from pl1 onto the end of pl2. (In both levels).
7. Think of the list ADT modified using the following strategy. Whenever an element is located using the isPresent() operation, that particular element is deleted from the current position and reinserted at the beginning of the list. The motivation behind this relocation is that in many situations an element accessed in a list is expected with high probability to be accessed several times in the future. So, keeping the element near the beginning of the list reduces average search time. Modify the list ADT implementations to incorporate this modification.
8. It is required to keep the data of employees in a general list. Write the type definition Entry such that each employee will have the following data (Name -Home Address -Date of Birth -Company (Name, Address, Phone number)), Note that, any date should include day, month, and year; any address should include street, city, and zip code.