|  |  |
| --- | --- |
| **Lab A** | **APPLICATIONS OF LINKED LIST** |
|  | **5-Sept ----10-Sept** |
| 1. Create a link list of user supplied ten characters to store a name. Create a second link list of same type of user supplied five characters. Now using a function remove (..), traverse first link list and if any three consecutive characters of second link list appears as consecutive characters of first link list, remove those from first link list. 2. Create a circular integer linked list. If the linked-list contains values less than 7 nodes, concatenate 0's at the second last positions. If the linked list has more number of elements, truncate it such that apart from the first and the last values, the first 6 values are displayed. 3. Write a program using linked list to reverse a linked list. 4. Write a program to implement multiplication of two polynomials. 5. Write a function to delete a node from a circular linked list. 6. Assume there are duplicate elements in a linked list. Sort the elements of the list and then print all duplicate elements from the list. 7. Write a function check whether a string in linked list is a palindrome or not. | |

|  |  |
| --- | --- |
| **Lab B** | **STACK AND QUEUE** |
| 1. For a given queue, Q, write a recursive function to shuffle the values stored in first half of queue with the second half of it in an alternate manner. The first and the last values of the queue are not affected by the shuffle. For example, (LHS is front of queue and RHS as rear of queue)   Example 1: Input Q: 1 2 3 4 5 6  Output Q: 1 5 3 4 2 6  Example 2: Input Q: 7 4 6 2 1  Output Q: 7 2 6 4 1   1. Write a program that reads integers from a file and pushes them into stack until it reads a negative no. Then it pops five items from the stack and prints them. If there are fewer than five items in the stack, print the error message. After printing the data, the program resumes reading data and placing them in the stack. When the end of file is detected, print a message and print items remaining in the stack. 2. Take 'n' number of strings from the user.   a. Sort them in alphabetical order. Use recursion function.  b. Reverse the sorting order and display.   1. Use Linked List: Given a queue, shuffle values of the queue, such that the smallest element is shifted to the end of the queue without changing the basic ordering of the elements in the queue. Further, write a modified stack for the same function such that the smallest element is shifted to the top of the stack without changing the ordering of the other elements. | |