**Assignment on**

**Data Structure and Algorithm Lab**

|  |  |
| --- | --- |
| **Statements** | |
| Exp. 01: | Write a program to create a dynamic linked list and perform the following operations on the list.   * Traversing the list just simply printing the values of the list * Insert a new node at the beginning, end, or middle of the list * Delete a given node from the list |
| Exp. 02: | You are to create a program that implements both a stack and a queue, and processes commands as follows: [**Handling Queue and Stack]**   * s-set mode to “STACK MODE” and (always) print the current contents of the stack on one line, separated by spaces, with the **top of the stack** at the left (don’t show the sentinel). * q-set mode to “QUEUE MODE” and (always) print the current contents of the queue on one line, separated by spaces, with the **head of the queue** at the left (don’t show the sentinel). * Any legal integer-push onto the stack (STACK MODE) or insert at the tail of the queue (QUEUE MODE) and print it. * p-either pop the top of the stack (STACK MODE) or remove the item at the head of the queue (QUEUE MODE) and print it. * Q-exit the program (be sure to free all memory) |
| Exp. 03: | Write a program that stores *n* numbers in an array, and sorts them by using the quick sort algorithm. [**Application of Stack]** |
| Exp. 04\*: | The Tower of Hanoi puzzle was invented by the French mathematician Edouard Lucas in 1883. You are given a tower of 8 disks (the picture below just shows 4 disks for the interest of spaces), initially stacked in decreasing size on one of the three pegs. Write a program to transfer the entire tower to one of the other pegs (the third one in the picture below), moving only one disk at a time, and never a larger one onto a smaller. **[Application of Recursion]**  Problem 2 Statement: The Tower of Hanoi puzzle was invented by the French mathematician Edouard Lucas in 1883. We are given a tower of eight disks The picture below just shows four disks for the interest of space), initially stacked in decreasing size on one of three pegs. The objective is to transfer the entire tower to one of the other pegs (the third one in the picture below), moving only one disk at a time and never a larger one onto a smaller. Lets call these 8 disks l, 2, 3, 4, to 8, 8 being the largest disk and l being the smallest. Lets call the pegs A, B, C. Design an algorithm to produce one solution for these 8 disks. Then output the sequence of disk movement. For instance, the correct movement sequence for 3 disks should move disk 1 from peg A to peg C move disk 2 from peg A to peg B move disk 1 from peg C to peg B move disk 3 from peg A to peg C move disk 1 from peg B to peg A move disk 2 from peg B to peg C move disk from peg A to peg C  \* The problem can be solved recursively. |