1806ICT Programming Fundamentals

Week 10 : Stacks, Queues, Linked Lists

1. In the Queue lecture, the ADT queue was implemented with an array. Now, implement the ADT queue using a linked list.

2. Using the supplied program (inFixtopostFix.c) which performs an in-fix to post-fix conversion of an arithmetic expression, add the % operator to the list of valid operators.

3. Using the supplied program (inFixtopostFix.c) which performs an in-fix to post-fix conversion of an arithmetic expression, implement a program which evaluates arithmetic expressions using the C precedence. The operands will be single character numbers and the possible operators are: ^, *, /, +, -, (,).

4. In the lecture slides on Linked Lists, the code for the function `insertNode()` was given. This function inserts a new node into the linked list at the position where the next node has a higher value than the new node. This is in effect, inserting the new node into the linked list in a sorted manner.

   Now, write another insertion function that inserts the node at the first position in the linked list following a node that stores a particular value. For example, if the linked list contains the values 1 5 7 8 10, and the new node to be inserted has the value 9 and it is to be inserted after the value "1", the new insertion function will cause the new node to be inserted in such a way that the resultant linked list will contain the values 1 9 5 7 8 10.

   You should use the function `findNode()` to help you find the node that stores a particular value. If the node is not present, the insertion should occur at the end of the list.

   Test the new insertion function with the rest of the functions given in the lecture, and the new functions written in Q1 and Q2 above.

5. Write a function `delDuplicate()` that deletes duplicate valued nodes in the linked list.

   Test the new delete function with the rest of the functions given in the linked list lecture, and the new functions written in Q1, Q2, and Q3 above.

6. Using a linked list, write a program that reads in a sequence of integer numbers and stores each number as a node in the linked list. The numbers do not need to be sorted in the list. Use the following structure for the nodes:

```
struct Node
{
      int data;
      struct Node *next;
};
typedef struct Node NODE;
```

Using the linked list that you created, your program will compute and print out the sum of all the integer numbers in the list, and print out the maximum and minimum numbers in the linked list.

Sample Run:

Input	Output
1 8 4	Sum=13, Max=8, Min=1
4 2 13 8 6	Sum=33, Max=13, Min=2