**Assignment – 7**

1. Write a program to convert a given infix expression into postfix notation.
2. Write a program to convert a given infix expression into prefix notation.
3. Write a program to covert a given postfix into infix expression.
4. Write a program to convert a given prefix into infix expression.
5. Write a program evaluate\_Polish which evaluates a reverse Polish expression (read about this notation if you don’t already know) from the command line using by using stacks, where each operator or operand is a separate argument. For example:

evaluate\_Polish 3 5 6 + \*

evaluates to

3 \* (5 + 6)

1. Write a program to reverse the order of words in a sentence by using a stack.

Example:

data structures and algorithms

on reversing the order of words becomes

algorithms and structures data

1. Assuming a stack of unique integers, write a program to sort it in ascending order by using recursion. Use push, pop, isempty, and isfull functions of stack.
2. Implement stack of unique integers that supports min operation. The min function returns the current minimum of the numbers in the stack, and can be called any number of times.
3. Write a program to implement a circular queue using an array.
4. Given two Linked Lists, create union and intersection lists that contain union and intersection of the elements present in the given lists. Order of elements in output lists doesn’t matter.

*Input: (can be directly given in program)*

List1: 10 15 4 20

List2: 8 4 2 10

*Output:*

List1: 10->15->4->20

List2: 8->4->2->10

Intersection List: 4->10

Union List: 10->15->4->20->8->2

1. Write a program to find the middle node of a singly linked list which contains only numbers. The input and output should be like the following.

*Input:*

10 15 4 20 8 4 12 11 7

*Output:*

List: 10->15->4->20->8->4->12->11->7

Middle Node: 8

(The numbers for the linked list should be given at run time. The output should display the linked list

and the middle node.)

1. Write a program, with functions that perform the following for a Singly linked list containing only integers.
   1. The smallest element
   2. The largest element

*Input:*

10 15 4 20 8 4 12 11 7

*Output:*

List: 10->15->4->20->8->4->12->11->7

Smallest element: 2

Largest element: 20

1. Write a program to implement a stack using linked list.
2. Write a program to implement a queue using linked list.
3. Write a function to reverse a linked list of integers (use pointer to a pointer as function arg). The program will take any number of nodes during run time and reverse the list (without counting the number of nodes in the list). The result should be the original list reversed, so the head becomes the last node and the last node becomes the head of the linked list. Print the original linked list and the reversed linked list.
4. Write a program to find if a linked list contains a loop without knowing the number of nodes. Use two pointers to traverse the list with different ‘speed’.
5. Write a Program to implement the rat in a maze game. Assume there is only one entry and one exit. There are blocks in its path and it can easily take a wrong route. Find a path from the entry to the exit.(Note: The maze can be viewed as a two dimensional array with 1's in paths that can be taken and 0's in paths that cannot be taken.)
6. Write a program to implement Queue Functions Using Arrays and Macros
7. [Write a Program to Implement various Queue Functions using Dynamic Memory Allocation](http://www.sanfoundry.com/c-program-implement-queue-functions/)
8. Write a program to solve the following using stack. The objective of the puzzle is to move the entire stack to another rod, obeying the following rules:

1) Only one disk must be moved at a time.

2) Each move consists of taking the upper disk from one of the rods and sliding it onto another rod, on top of the other disks that may already be present on that rod.

3) No disk may be placed on top of a smaller disk.