Lab 3. Stacks and Queues

Problem 1. Evaluation of expression:

Infix, postfix, and prefix notations are different but equivalent notations of writing algebraic expressions. For example, (A + B) * C (Infix expression) can be written as:

- *+ABC in the prefix notation.

- AB+C* in the postfix notation.

Write a program to do the following tasks:

a) Implement a stack using a linked list.
b) Convert an infix expression into its equivalent postfix notation.
c) Evaluate the postfix expression.
d) Compute the value of the postfix expression.

Problem 2. Parentheses Checker:

An algebraic expression is valid if for every open bracket there is a corresponding closing bracket. For example, the expression (A+B} is invalid but an expression {A + (B – C)} is valid.

Write a program to check nesting of parentheses by stack.

Problem 3. In the Josephus problem, n people stand in a circle waiting to be executed. The counting starts at some point in the circle and proceeds in a specifc direction around the circle. In each step, a certain number of people are skipped and the next person is executed (or eliminated). The elimination of people makes the circle smaller and smaller. At the last step, only one person remains who is declared the 'winner'. Therefore, if there are n number of people and a number k which indicates that k–1 people are skipped and k–th person in the circle is eliminated, then the problem is to choose a position in the initial circle so that the given person becomes the winner.

Write a program to do the following tasks:

a) Implement a queue.
b) Find the solution of the Josephus problem using queue.