BITS Pilani
Hyderabad Campus

First Semester 2023-24
Data Structures and Algorithms Design (Merged-SEZG519/SSZG519)
Exercises (Elementary Data Structures)

1. Convert the following Infix expressions to Prefix and Postfix expressions.
   a. ((L+(M*N))/(O-P))
      i. Prefix: /+L*MN-OP
      ii. Postfix: LMN*+OP-/
   b. ((L+M)*(N+P))
      i. Prefix: LM+NP+*
      ii. Postfix: *+LM+NP
   c. (L+(M*N))
      i. Prefix: LMN*+
      ii. Postfix: +L*MN
   d. (L*(M*(((N+L)+M)*N)))
      i. Prefix: LMNL+M+N***
      ii. Postfix: *L*M*++NLMN
   e. ((H*((((L+((M+N)*O))*F)*G)*P))+J)
      i. Prefix: +*H***+L*+MNOFGPJ
      ii. Postfix: HLMN+O*+F*G*P**J+
2. The following algorithm is to implement the stack using two queues (i.e., Q1 and Q2) where pop and tos algorithms are computationally costly. Re-write push, pop, and tos algorithms where the computation complexity of push algorithm is high.

```
Algorithm push(o):
        if Q1.size( ) = N then
                indicate that a stack-full error has occurred
        Q1.enqueue(o)
        for i = 1 to Q1.size( )-1 do
            Q2.enqueue(Q1.dequeue( ))
        Q1.enqueue(Q1.dequeue( ))
        for i = 1 to Q2.size( ) do
            Q1.enqueue(Q2.dequeue( ))
Algorithm pop( ):
        if Q1.isEmpty( ) then
                indicate that a stack-empty error has occurred
        e ← Q1.dequeue( )
        return e
Algorithm tos( ):
        if Q1.isEmpty( ) then
                indicate that a stack-empty error has occurred
        return Q1[f]
```

3. The following algorithm is to implement the stack using two queues (i.e., S1 and S2) where dequeue algorithms is computationally costly. Re-write enqueue and dequeue algorithms where the computation complexity of enqueue algorithm is high.

```
Algorithm enqueue(o):
        if S1.size( ) = N then
                indicate that a queue-full error has occurred
                return
        for i = 1 to S1.size( ) do
            S2.push(S1.pop( ))
        S1.push(o)
        for i = 1 to S2.size( ) do
            S1.push(S2.pop( ))
Algorithm dequeue( ):
        if S1.isEmpty( ) then
                indicate that a queue-empty error has occurred
                return NULL
        e ← S1.pop( )
        return e
```