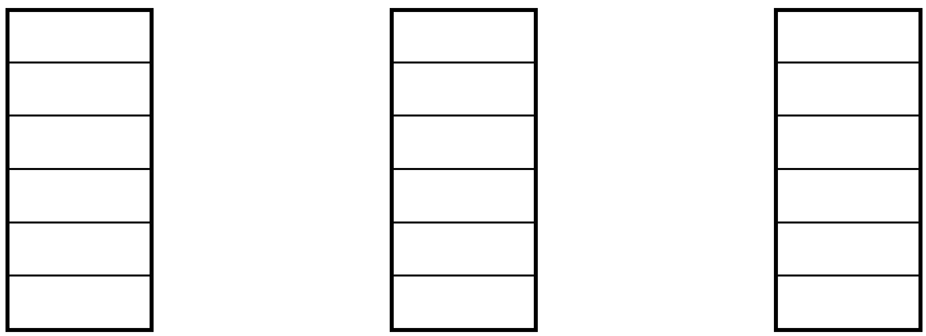
Computer Science I Worksheet: Stacks and Queues

1) Convert the following infix expression into its equivalent postfix expression using a stack. Show the contents of the operator stack at the indicated points in the infix expressions (points 1, 2 and 3), and also the final postfix expression. You may draw another stack alongside for your work.

```
                          1                    2              3
A  *  (  B  +  C    /    ( D + E ) )  –   F  * ( G  /   H  +  I )
```



1 **2** **3**

postfix expression upto 1 : ____

postfix expression upto 2 : ____

postfix expression upto 3 : ____

Final postfix expression : ____

2) Imagine implementing a Double Ended Queue (DEQ). A DEQ allows for enqueues from the front AND back and dequeues from the front AND back. (Thus, we would have to add two operations to a DEQ: enqueues to the front and dequeues from the back.)

a) Given the partial implementation posted online, complete the implementation for the function enqueueFront within the framework provided below:

```
int enqueueFront(struct queue* qPtr, int val) {

    struct node* temp;
    temp = (struct node*)malloc(sizeof(struct node));

    // Fill in code inside the if clause.
    if (temp != NULL) {









        return 1;
    }
    else
        return 0;
}
```

b) Explain why within this framework an efficient implementation of dequeueBack can not be accomplished. What changes would need to be made to the struct to allow for an efficient implementation of dequeueBack?