Lab 8 Tasks

Your task is to implement two of the most commonly used data structures: stack and queue using **linked list**. For both tasks create and use a structure that has an integer variable and a pointer of that structure type [To point to the next element in the list].

Q1. Queue is FIFO(First In First Out) data structure. Here the element which is added first will be removed first. Figure 1 illustrates a typical queue.

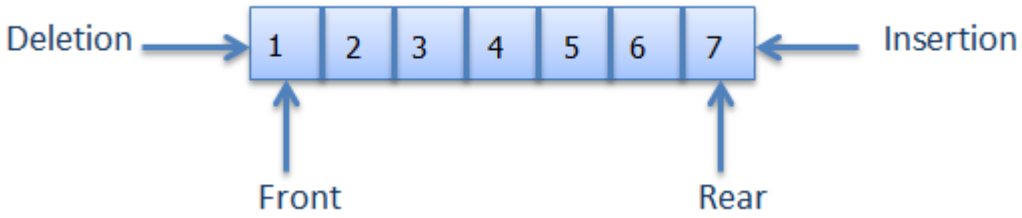

Figure 1: A typical queue

When implementing a queue, a pointer will point to the first element in the queue [head] and another pointer will point to the last element in the queue [tail]. Declare the pointers in the main function. Write two separate functions called *enqueu*() and *dequeue*():

i. *enqueue():* The job of this function is to insert a new element at the rear of the queue. It will take two values as parameter. The first will be the pointer to the last element in the queue. The other one will be an integer value that is to be added to the queue.

ii. *dequeue():* This function will the remove a single element from the front of the queue and return the value stored in that element. It will only take a pointer to the first element in the queue as a parameter.

Q2. Stack is LIFO (Last In First Out) data structure. Here the element which is added at last will be removed first. Figure 2 illustrates a stack.

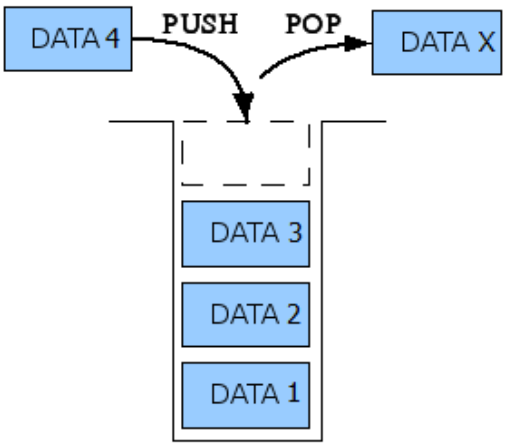

Figure 2: A stack

When implementing a stack, a pointer will point to the first element in the stack [head]. Declare the pointer in the main function. Write two separate functions called *push*() and *pop*():

i. *push():* The job of this function is to insert a new element in the stack. It takes two parameters. The first one is a pointer to the element at the top of the stack. The other one will be an integer value that is to be added to the stack.

ii. *pop():* This function will the remove a single element from the top of the stack and return the value stored in that element. It will only take a pointer to the top element in the stack as a parameter.