

# CCT A3 - The Circular Queue

A circular queue is a data structure based on the concept FIFO - First In First Out, just like a queue with its ends connected.

In the last problem you implemented a normal linear queue.

Your task now is to implement a circular queue using a circular linked list.

Your implementation must include the following functionalities :

queueNode\* head; - element to the head of the queue

void push(int element) - push element to the back of the queue

void pop() - remove one element from front of queue

int empty() - returns 1 if queue is empty and zero otherwise

You are NOT allowed to add global variables/arrays to the body of the code.

All the functions must have an  $O(1)$  running complexity.

You are not responsible for taking any input or output.

You are not supposed to modify any part of the code except the BODY of the code.

Only edit the body of the code and implement the required functions with the given prototypes.

NOTE : Also read about the advantages and disadvantages of using a circular linked list instead of an array for representing a Queue.

## INPUT

Not Needed

## OUTPUT

Not Needed

### Sample Input 0

```
10
push 1
push 2
push 3
push 4
pop
top
pop
top
pop
top
```

### Sample Output 0

```
2
3
4
```

