

## Lab 5: Circular Queue ADT

Implement Circular Queue ADT using an array. Your Circular Queue ADT has two main operations, Enqueue and Dequeue. In addition, you should implement two more functions for printing the first element and the last element.


- **Enqueue** a new element at the end of the element in the queue. If your queue is full, just print an error message.
- **Dequeue** the node in the front. If your list does not have any element, just print an error message.
- **PrintFirstElem** print the first element in the queue. If your queue is empty, just print an error message.
- **PrintLastElem** print the last element in the queue. If your queue is empty, just print an error message.

### 1. Input

Obtain a list of operations from the given input file, and execute the given operations in order. A detailed specification of the operations is provided below. Each line represents a single operation. Each operation and the necessary parameters are separated by a space. You may assume that the input values (represented as x and y below) are any characters. Note that max\_queue\_size is the number of elements that can be stored in the queue.

- **e x**: enqueue a new element with the key “x” after the last element
- **d** : dequeue the first element in the queue
- **f** : print the first element in the queue
- **r**: print the last element in the queue
- **n x**: create a new queue with the size of x. The number x is the maximum size of the queue. This operation will always be given in the first line of the operations in your input file.

An input file is shown below.



```
n 5
f
r
e 13
d
d
e 10
e 3
f
r
e 1
e 5
e 6
e 15
f
r
```

## 2. CircularQueue ADT

### (1) Data Specification for the objects

```
struct CircularQueue {  
    int *key;  
    int first;  
    int rear;  
    int qsize;  
    int max_queue_size;  
};
```

### (2) Function specification

- CircularQueue MakeEmpty( int max );
- int IsEmpty( CircularQueue Q );
- int IsFull( CircularQueue Q );
- void Dequeue( CircularQueue Q );
- void Enqueue( CircularQueue Q, char X );
- void PrintFirst (CircularQueue Q);
- void PrintRear (CircularQueue Q);
- void DeleteQueue (CircularQueue Q );

### 3. Program description

- name : p5.c
- input : a list of operations in a file (an input file name is given as a command line argument. See an example in “1. input” on the first page)
- output : the corresponding result in the standard output

Submit to the course website (<https://portal.hanyang.ac.kr>) your source code and a written report. Your report should include the description of your own implementation.