LAB5

Data Structure

Lab5

- The deadline for lab5 submission is 5th April at 11:59 pm.

- Folder name : lab5
- Code name: p5.c
- Each code will be tested by 5 different input files.
- 20 score for each input, if you don’t get the answer, you get 0 score.

Evaluation criteria

Category	Evaluation	
p5	100	
Total	100	

- *Use GCC **11** version.*
- *No score will be given if the gcc version is different.*

Lab5 – CircularQueue

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

Lab 5 – CircularQueue

- **n x** create a new queue with the size of x. The number x is the maximum size of 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

Lab 5 – CircularQueue (by array implementation)

- Structure

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

- Function

```
<Lab5>
CircularQueue MakeEmpty(int max);
int IsEmpty(CircularQueue Q);
int IsFull( CircularQueue Q );
void Dequeue( CircularQueue Q );
void Enqueue( CircularQueue Q, int X );
void PrintFirst (CircularQueue Q);
void PrintRear (CircularQueue Q);
void DeleteQueue (CircularQueue Q );
```

Lab 5 – CircularQueue

- input file : input.txt

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

- Result

```
Queue is Empty!
Queue is Empty!
Enqueue 13
Dequeue 13
Dequeue failed : Queue is Empty!
Enqueue 10
Enqueue 3
First : 10
Rear : 3
Enqueue 1
Enqueue 5
Enqueue 6
Enqueue failed : Queue is Full!
First : 10
Rear : 6
```

Lab 5 – CircularQueue

```h
#include<stdio.h>
#include<stdlib.h>

struct CircularQueueStruct{
        int *key;
        int first;
        int rear;
        int qsize;
        int max_queue_size;
};

typedef struct CircularQueueStruct* CircularQueue;
```

```c
void main(int argc, char* argv[])
{
        char command;
        FILE *input;
        CircularQueue queue;
        int queueSize;
        int tmpNum;

        input = fopen(argv[1], "r");
        //queue = MakeEmpty(queueSize);

        while(1) {
                command = fgetc(input);
                if(feof(input)) break;
                switch(command) {
                        case 'n':
                                fscanf(input, "%d", &queueSize);
                                queue = MakeEmpty(queueSize);
                                break;
                        case 'e':
                                fscanf(input, "%d", &tmpNum);
                                Enqueue(queue, tmpNum);
                                break;
                        case 'd':
                                Dequeue(queue);
                                break;
                        case 'f':
                                PrintFirst(queue);
                                break;
                        case 'r':
                                PrintRear(queue);
                                break;
                        default:
                                break;
                }
        }

        DeleteQueue(queue);
}
```

Lab 5 – Queue

- program name : p5.c
- input : a list of operations in a file.
- output : the corresponding result in the standard output.