

# Queues – Implementations

In this lab, we will implement queues of integers in different ways and then use the different implementations interchangeably. The interface we'll be using is the following :

```
#ifndef __QUEUE__H__
#define __QUEUE__H__

void init_queue(queue_t*); // initialize the data structure
int push_queue(queue_t *, int); // push value into queue
int pop_queue(queue_t *, int *) // pops value from queue;
void display_queue(queue_t *); // displays the contents of the queue

#endif
```

## Ex 1 - Array Implementation

Use an array of MAX entries in this implementation of the queue. the following part of the header file for this implementation

```
#define MAX 100
```

Write a recursive version of the display\_queue function.

## Ex 2 – Linked List Implementation

Define a node type and then do the implementation of the interface. Write an iterative and recursive version of the display function.

### Ex 3 – Queue Usage – Simulations

In this exercise, we're going to use queues in order to simulate scheduling in a doctor's waiting room. For that purpose, we model patients in the following manner :

**table will be given in class**

A patient has an arrival time and stays in the waiting room until he's received by the doctor to be examined. At some point, the doctor is going to send him/her to do some more examinations (blood, scanner, ...). In the mean time, the doctor examines another patient. When the patient returns from the external examination, he reenters the waiting room and awaits his turn again !

- 1.** Do the simulation of such a system given that a patient is chosen from the waiting room in a first-come first-served basis. Calculate the average global time of patients as well as the average waiting time.
- 2.** Do the simulation given that a patient is chosen based on the duration of the examination instead of his arrival time. Compare both versions of the simulation using the average global time and the average waiting time.