1.What differences have you found between using stacks or using queues to traverse a graph looking for a connection or not between two nodes?

Truly there is not a huge different into use stack or queues because you are doing the same thing but with a different data structure , the only difference can be the implementation of insert and extract something in the data structure . If you use a stack when you insert something and then you have to extract you know that is a LIFO (last in first out) so the last thing you have pushed is the first that you have to pop and in the queue is different as it is a FIFO (first in first out) so the first element that you pushed is the first to be poped. At the end, the exercise could be implmented just replacing the old stack functions by the new queue ones and doing some minor changes.

2. Regarding Exercise 3, if it had not been necessary to rename the files to deliver the practices, assuming that you could have overwritten the files that were already in previous Exercises (queue.h, queue.c, p3_testqueue.c, .. .):

a.What files have had to be modified in Exercise 3a to move from the implementation of queues with the array to the implementation based on lists? What has been changed and why?

The exercise 3a ask us to make the implementation of a queue using a circular list. We did so by modifying the queue.c and adapting the logic of the queue functions to work with the new functions of list.c.

The first change was the structure of the queue. In previous exercises the structure had one void pointer for items and two integers for determining the front and rear position in the queue but in exercise 3 this changed because we have to implement a queue using list, so this queue doesn´t needs any integer for know the position or a void * for having the information about the things that contains the queue. This new implementation only needs a pointer to a list that contains in its structure a pointer to a NodeList that has the information of the node and its next linked node.

Then we changed all the primitives function for readapt the code for the new struct,.
In queue_ini we changed the initialization of the values for readapt to the new structure and called list_ini(), in queue_destroy we called the function list_destroy , in the empty and full function we check the list instead of seing the positions of the queue extrems, in the functions of extract and insert we called the functions insert_last and extract_first instead of using the integer values of front and rear that we used in the previous part, and we do the same thing for the rest of the functions, calling to the primitives of list.c.

All of this has been changed for the implementation of the queue with a circular linked list this is the reason why we changed the structure and we used a pointer to a list , also we can see this in parts of the code like the implementations of the functions extract and insert of the queue.c . In those functions we call to the functions extract_first (on extract) and insert_last (on insert). Those functions have been implemented in list.c to make something similar as a queue that consist in a FIFO (first in , first out) so the first thing that you introduce “insert_last at the end of the list” is the first thing that you extract “extract_first extract something of the begging of the list”.

b. What files have had to be modified in Exercise 3b (with respect to Exercise 1b) to now use the queues implemented on lists? What has been changed and why?

In exercise 3b of the practice we have to test the queuel with the p3_testqueue.c we did not have to change a lot this file because this program only call the functions of queue , that means that as we change the implementation of the primitives of queue in the queuel.c but we don’t change the .h file so we use the same protocol functions we don´t have to change a lot of thing of the file that was

provided by the teachers of this lesson , the only thing that we changed was the name of the include in which we put queuel.h instead of queue.h in order to avoid library linking errors.