**Brief Report**

Our code consists of four source files, three header files. In the Simulation Engine and queues part, we put them into separate files with corresponding header file associated with it.

In the Priority Queue part, we define three functions: one is to create a new queue, another one is to insert new structure into an existing queue(create one if not exists), order them according to their timestamp, and the last one is about pop out the items with the highest priority.

In the FIFO Queue part, very similar to Priority Queue, we define three similar functions, but it will always pop out the items that first get into the queue and insert the items behind the items we inserted before.

In the Simulation Engine part, runsim is created, which is a switch for the simulation. Once we call this function in the main function, the simulation begins. We use while loops to simulate all the events in the list and print out the airplane id and its corresponding arrival and landing time.

At last, in application part, we define Arrival functions, Landing functions, callback function to call arrival and landing functions, randexp function to generate time, In the main, we initialize the first entity of the simulation, and run the simulation.

**Evidence for running correctly**

One could see our output runs; Every time we get different results, Landing event for a specific airplane comes after the arrival event. Airplane’s id is in the sequential order. Events are arranged in a reasonable way. There are some chances that airplanes’ arrival events and landing events happen at the same time. The simulation results are very close to the reality, which could be the evidence.