```
struct Plane
        ID, arrival_time, lock, cond
end struct

struct Queue
        capacity, size, front, rear, element_array
end struct

// declare landing, departing, and emergency queues based on struct Queue

// declare necessary mutex lock and condition variables

// you can also declare other necessary variable, like locks, conds, or other utility functions for
enqueueing, dequeueing, initializations, etc

FUNCTION landing_func
        // declares a struct Plane
        // generates a unique plane id
        // generates arrival time
        // initializes the lock attribute of struct Plane
        // initializes the cond attribute of struct Plane
        If it is an emergency landing
                enqueue the plane to emergency queue
        else
                enqueue the plane to landing queue
        if this is the first plane
                notify the ATC thread
        // wait for signal from air traffic controller
        // pthread_exit
END FUNCTION

FUNCTION departing_func
        // declares a struct Plane
        // generates a unique plane id
        // generates arrival time
        // initializes the lock attribute of struct Plane
        // initializes the cond attribute of struct Plane
        // enqueue the plane to departing queue
        if this is the first plane
                notify the ATC thread
        // wait for signal from air traffic controller
        // pthread_exit
END FUNCTION
```

```
FUNCTION air_traffic_control
        // wait for signal from the first plane
        while current_time < start_time + simulation_duration
                if number of planes in emergency queue >  0
                        land the front plane in the emergency queue
                        pthread_sleep(t)
                else
                        if no starvation in landing queue and number of planes in departing queue
        >  0 and ( the front plane in the departing queue has been waiting longer than wait
        threshold or number of planes in departing queue > threshold or landing queue is empty)
                                depart the front plane in departing queue
                                pthread_sleep(t)
                        else
                                land the front plane in landing queue
                                pthread_sleep(t)
        // pthread_exit
END FUNCTION

FUNCTION main
        // parse command line parameters
        // initialize mutexes, conds, and queues
        // create air_traffic_control_thread
        // create one plane thread that runs landing function
        // create one plane thread that runs taking off function
        while current_time < start_time + simulation_time
                rand = generate_random_number()
                If rand <= p || time for an emergency landing plane
                        // create one plane thread running landing function
                If rand <= 1 - p
                        // create one plane thread running taking off function
                pthread_sleep(1)
        end while
END FUNCTION
```