Threads(Airline Reservation System)

We have two thread functions. They both take the pointer parameter and pass it to the specific seat. Inside the functions we have a while loop which makes sure that we are not doing anything when the plane ends up full. Then we set the seed as NULL equivalent to the computer's clock and which generates different sequences each time. After that there is a while loop that checks whether it is the thread's time to run or not with the information that says the plane is not full yet. Under that we declare a row and column variable and generate a random value for each. Several printf statements and converting the row and column into a particular format. Then passing parameters for thread function to seat information in order to mark who owns the seat.
ATTENTION !! After the while loop I put a condition for reserve info in order to achieve simplicity.
I already know that thread does not necessarily require to make reservation however if it directly passes its turn to other thread runtime grows enormously and for complexity issues I stick onto conditions that make reservation for the thread. (Also I do not assert it is the best way, you can assume I'm not capable of implementing this as I'm an average CS student with 6 courses.)
For the main thread actually there is nothing too complex. First we declare threads and integers for passing as an argument to thread functions. Then we create the threads with a start argument at the end with the integer values that we just declared. However, the most vital part here is we need to keep track of the plane seats. I implemented a for loop for that. It goes through all the seat values and if there is not any seat left it tells that the plane is full now.
Then there is nothing fancy. We join our threads in sync. After that there comes the classical output part. We go through every value of the vector matrix and print out the value for seat plan.