Parampreet Grewal Singh

1. How many threads are you going to use? Specify the task that you intend each thread to perform.

I will be using three threads, one for the main thread, one for the customer, one for the clerks

2. Do the threads work independently? Or, is there an overall "controller" thread?

The threads can work together, for example, the main thread manages the customer and clerk threads that will be used

3. How many mutexes are you going to use? Specify the operation that each mutex will guard.

There will be a queue mutexes , one for economy and one for business class queue
There will be a clerk mutex, five for each clerk
There will also be a statistic mutex, one mutex for shared statistics

4. Will the main thread be idle? If not, what will it be doing?

The main thread will not be idle, it will be reading the input file, creating the threads and managing them.

5. How are you going to represent customers? what type of data structure will you use?

The customer structure will contain their ID, the class type, arrival time, and the time it takes to service them
I will be using a FIFO queue data structure, one queue for the economy customers and another for the business customers

6. How are you going to ensure that data structures in your program will not be modified concurrently?

Use the mutexes to ensure they wont be modified concurrently

7. How many convars are you going to use? For each convar: (a) Describe the condition that the convar will represent. (b) Which mutex is associated with the convar? Why? (c) What operation should be performed once pthread cond wait() has been unblocked and reacquired the mutex?

a) For economy queue, it signals addition to the economy queue
   For business queue, it signals addition to the business queue
   For clerks, it signals when a clerk is available
b) Each convar will be associated with an economy/business queue or a clerk mutex
c) We would need to recheck the economy/business queue or clerk status

8. Briefly sketch the overall algorithm you will use. You may use sentences such as: If clerk i finishes service, release clerkImutex.

```
Initialize all mutexes and condition variables

Read customer information from "customer_info.txt" file
For each customer entry in the file:
    Parse the entry and store it in the customer_list array

Create clerk threads
    Create a thread with clerk_entry function and pass clerk ID

Create customer threads
```

```
    Create a thread with customer_entry function and pass customer info

Wait for all customer threads to terminate
    Join the customer thread

Release all mutexes and condition variables

Calculate and print the average waiting time of all customers
```