1. How many threads are you going to use? Specify the task that you intend each thread to perform.
A thread will be created for each flow that is in the input file. Each thread will sleep until it's "arrival time" comes. Then the thread will "transmit" until the specified time is achieved.

2. Do the threads work independently? Or, is there an overall “controller” thread?
The threads work independently and they will sort the queue them selves.

3. How many mutexes are you going to use? Specify the operation that each mutex will guard.
There will be one mutex per thread and it will guard the queue sorting operation.

4. Will the main thread be idle? If not, what will it be doing?
The main thread will be waiting for all the threads to terminate.

5. How are you going to represent flows? What type of data structure will you use?
A structure that conatins arrival time, transmission time, priority and id(Flow No).

6. How are you going to ensure that data structures in your program will not be modified concurrently?
A new structure will be made for every thread.

7. How many convars are you going to use? For each convar:
(a) One conditional variable will be used to determine when the thread finishes transmission.
(b) All mutexs that are waiting in queue will be associated with that thread(c) Once the pthread_cond_wait has been unblocked then the acuired mutex should find the next thread in queue then start transmission.

8. In 25 lines or less, briefly sketch the overall algorithm you will use
Flow arrives and requests pipe, lock trans mutex.
If there is no transmission and queue is empty then start transmission, release trans mutex.
If there is a transmission or the queue is empty, add flow to queue then sort the queue.
After the queue is sorted wait for broadcast from transmission.
If signal arrives check if the flow is at the front of the queue.
If it is not, continue waiting, if it is then remove it from the top of the queue, release trans mutex and start transmission.