Lab Tasks:

Task1:

a. Create an icecream eating contest problem protected by a semaphore lock. Use global variables icecreamremaining. Create 3 threads for 3 persons to eat icecreams until all are finished. But only one person will be given icecreamcone at a time by the salesman. So restrict access to icecreams so that only one thread can decrement it by using a semaphor. So use sem_wait and Sem_post in thread .

b. get each person to count money from his wallet for the icecream payment which can take one to 2 seconds during which other person can acquire the salesman to sell him icecream. Redo the coding to accommodate this condition in your threads as well.

Task 2:

You need to synchronize customers at boarding lounge of an airport using semaphore. where there are
10 customers, each needs to weight his luggage, get it checked and get a boarding pass. During each
task passengers are too bored that they sleep, weighting luggage takes 4 seconds sleep, security
check for luggage needs 7 seconds sleep and getting boarding pass needs 3 seconds sleep.

Task 3:

Imagine you are designing a ticketing system for a popular concert. The system will have multiple ticket sellers, each running on a separate thread. To prevent overselling, you need to ensure that no two sellers can sell the same ticket at the same time. How would you use semaphores in C to ensure that only one ticket seller can access the ticket database at a time, while the others wait until the resource becomes available? (Create a code based implementation)