Modified Dining Philosopher's Problem

The dining philosophers problem contains five philosophers sitting on a round
the table can perform only one of two actions – eat and think.

In the first Part of the question, we have to create five threads and we have to run these five threads in a infinite loop

In this part we have written a code in which no Deadlock is happening also 1 philosopher

is eating at 1 time. In this we create a function eat pattern that decides which philosopher is eating and which completed its food.

In the part one of Question 1 we have create these 5 threads and run in a infinite loop
In this part we use semaphores in which 2 philosophers is eating the food when they both completes its food then next 2 philosophers repeats this process
We run in an infinite loop to ensure that there is no deadlock happens.
Utilization of semaphores to access the resources.

Also make a makefile so that so that when we sun makefile all the object files makes.