

12. Design a C program to simulate the concept of Dining-Philosophers problem.

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define NUM_PHILOSOPHERS 5

sem_t chopstick[NUM_PHILOSOPHERS];

void* philosopher(void* num) {
    int id = *(int*)num;
    printf("Philosopher %d is thinking...\n", id);
    sleep(1);

    // Pick up left chopstick
    sem_wait(&chopstick[id]);
    printf("Philosopher %d picked up left chopstick.\n", id);

    // Pick up right chopstick
    sem_wait(&chopstick[(id + 1) % NUM_PHILOSOPHERS]);
    printf("Philosopher %d picked up right chopstick and starts eating.\n", id);
    sleep(2); // Eating time

    // Put down chopsticks
    sem_post(&chopstick[id]);
    sem_post(&chopstick[(id + 1) % NUM_PHILOSOPHERS]);

    printf("Philosopher %d put down chopsticks and starts thinking again.\n", id);
    return NULL;
}
```

```

int main() {
    pthread_t threads[NUM_PHILOSOPHERS];
    int philosopher_ids[NUM_PHILOSOPHERS];

    // Initialize semaphores
    for (int i = 0; i < NUM_PHILOSOPHERS; i++)
        sem_init(&chopstick[i], 0, 1);

    // Create philosopher threads
    for (int i = 0; i < NUM_PHILOSOPHERS; i++) {
        philosopher_ids[i] = i;
        pthread_create(&threads[i], NULL, philosopher, &philosopher_ids[i]);
    }

    // Wait for all threads to complete
    for (int i = 0; i < NUM_PHILOSOPHERS; i++)
        pthread_join(threads[i], NULL);

    // Destroy semaphores
    for (int i = 0; i < NUM_PHILOSOPHERS; i++)
        sem_destroy(&chopstick[i]);

    return 0;
}

```

OUTPUT:

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define NUM_PHILOSOPHERS 5
sem_t chopstick[NUM_PHILOSOPHERS];
void* philosopher(void* num) {
    int id = *(int*)num;
    printf("Philosopher %d is thinking...\n", id);
    sleep(1);

    // Pick up left chopstick
    sem_wait(&chopstick[id]);
    printf("Philosopher %d picked up left chopstick.\n", id);

    // Pick up right chopstick
    sem_wait(&chopstick[(id + 1) % NUM_PHILOSOPHERS]);
    printf("Philosopher %d picked up right chopstick and starts eating.\n", id);
    sleep(2); // Eating time

    // Put down chopsticks
    sem_post(&chopstick[id]);
    sem_post(&chopstick[(id + 1) % NUM_PHILOSOPHERS]);
```

```
Philosopher 0 is thinking...
Philosopher 1 is thinking...
Philosopher 2 is thinking...
Philosopher 3 is thinking...
Philosopher 4 is thinking...
Philosopher 4 picked up left chopstick.
Philosopher 3 picked up left chopstick.
Philosopher 2 picked up left chopstick.
Philosopher 1 picked up left chopstick.
Philosopher 0 picked up left chopstick.
```