i. Dining Philosopher

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

#define N 5

sem_t forks[N];
sem_t room;

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

        sem_wait(&room);
        sem_wait(&forks[id]);
        sem_wait(&forks[(id + 1) % N]);

        printf("Philosopher %d is eating...\n", id);
        sleep(2);

        sem_post(&forks[id]);
        sem_post(&forks[(id + 1) % N]);
        sem_post(&room);
    }
}

int main() {
    pthread_t tid[N];
    int ids[N];

    sem_init(&room, 0, N - 1);
    for (int i = 0; i < N; i++)
        sem_init(&forks[i], 0, 1);

    for (int i = 0; i < N; i++) {
        ids[i] = i;
        pthread_create(&tid[i], NULL, philosopher, &ids[i]);
    }
```

```
    for (int i = 0; i < N; i++)
        pthread_join(tid[i], NULL);

    return 0;
}
```

```
Philosopher 0 is thinking...
Philosopher 1 is thinking...
Philosopher 2 is thinking...
Philosopher 3 is thinking...
Philosopher 4 is thinking...
Philosopher 0 is eating...
Philosopher 1 is eating...
Philosopher 2 is eating...
Philosopher 3 is eating...
Philosopher 4 is eating...
Philosopher 0 is thinking...
Philosopher 1 is thinking...
Philosopher 2 is thinking...
Philosopher 3 is thinking...
Philosopher 4 is thinking...
...
```