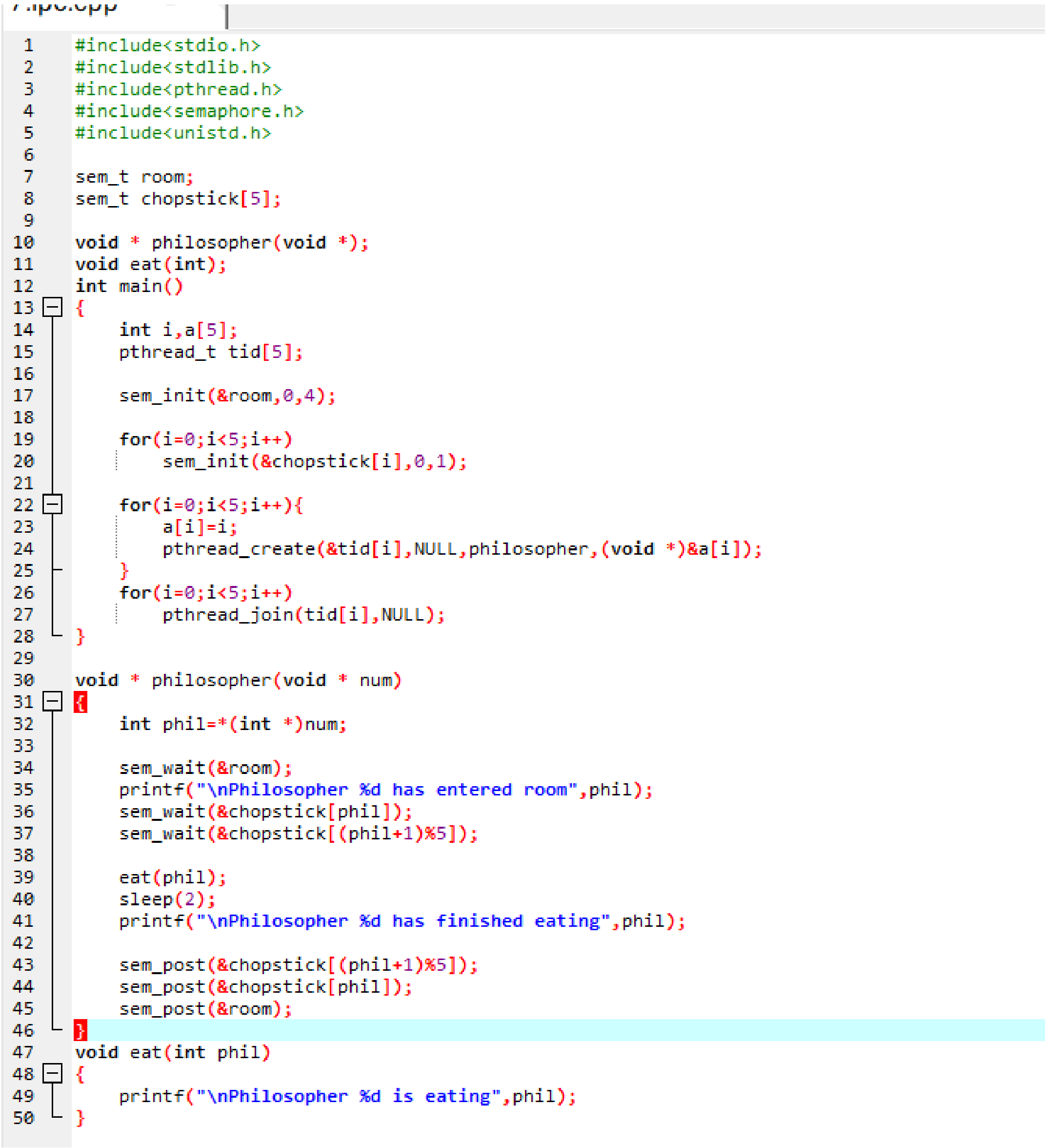
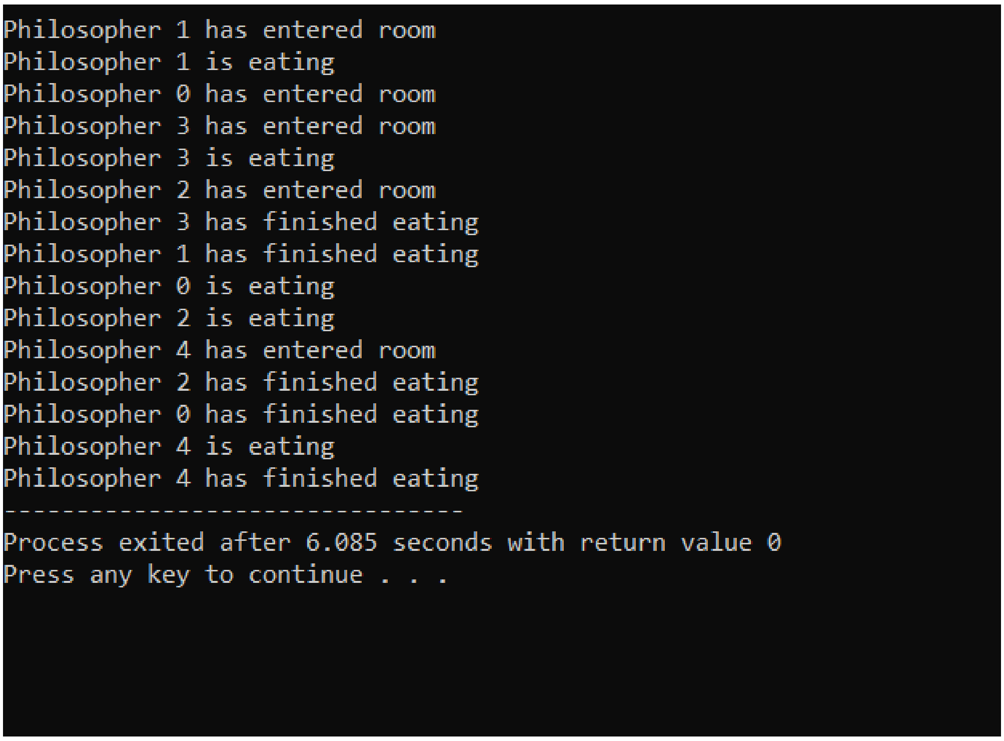
9. Design a C program to simulate the concept of Dining-Philosophers problem



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

sem_t room;
sem_t chopstick[5];

void * philosopher(void *);
void eat(int);
int main()
{
    int i,a[5];
    pthread_t tid[5];

    sem_init(&room,0,4);

    for(i=0;i<5;i++)
        sem_init(&chopstick[i],0,1);

    for(i=0;i<5;i++){
        a[i]=i;
        pthread_create(&tid[i],NULL,philosopher,(void *)&a[i]);
    }
    for(i=0;i<5;i++)
        pthread_join(tid[i],NULL);
}

void * philosopher(void * num)
{
    int phil=*(int *)num;

    sem_wait(&room);
    printf("\nPhilosopher %d has entered room",phil);
    sem_wait(&chopstick[phil]);
    sem_wait(&chopstick[(phil+1)%5]);

    eat(phil);
    sleep(2);
    printf("\nPhilosopher %d has finished eating",phil);

    sem_post(&chopstick[(phil+1)%5]);
    sem_post(&chopstick[phil]);
    sem_post(&room);
}
void eat(int phil)
{
    printf("\nPhilosopher %d is eating",phil);
}
```

Output:



```
Philosopher 1 has entered room
Philosopher 1 is eating
Philosopher 0 has entered room
Philosopher 3 has entered room
Philosopher 3 is eating
Philosopher 2 has entered room
Philosopher 3 has finished eating
Philosopher 1 has finished eating
Philosopher 0 is eating
Philosopher 2 is eating
Philosopher 4 has entered room
Philosopher 2 has finished eating
Philosopher 0 has finished eating
Philosopher 4 is eating
Philosopher 4 has finished eating
--------------------------------
Process exited after 6.085 seconds with return value 0
Press any key to continue . . .
```