```
1    #include<stdio.h>
2    #include<stdlib.h>
3    #include<pthread.h>
4    #include<semaphore.h>
5    #include<unistd.h>
6
7    sem_t room;
8    sem_t chopstick[5];
9
10   void * philosopher(void *);
11   void eat(int);
12   int main()
13   {
14       int i,a[5];
15       pthread_t tid[5];
16
17       sem_init(&room,0,4);
18
19       for(i=0;i<5;i++)
20           sem_init(&chopstick[i],0,1);
21
22       for(i=0;i<5;i++){
23           a[i]=i;
24           pthread_create(&tid[i],NULL,philosopher,(void *)&a[i]);
25       }
26       for(i=0;i<5;i++)
27           pthread_join(tid[i],NULL);
28   }
29
30   void * philosopher(void * num)
31   {
32       int phil=*(int *)num;
33
34       sem_wait(&room);
35       printf("\nPhilosopher %d has entered room",phil);
36       sem_wait(&chopstick[phil]);
37       sem_wait(&chopstick[(phil+1)%5]);
38
39       eat(phil);
40       sleep(2);
41       printf("\nPhilosopher %d has finished eating",phil);
42
```

```c
                     sem_init(&chopstick[1],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);
}
```

```
Philosopher 0 has entered room
Philosopher 0 is eating
Philosopher 2 has entered room
Philosopher 1 has entered room
Philosopher 3 has entered room
Philosopher 2 is eating
Philosopher 0 has finished eating
Philosopher 2 has finished eating
Philosopher 4 has entered room
Philosopher 3 is eating
Philosopher 1 is eating
Philosopher 3 has finished eating
Philosopher 1 has finished eating
Philosopher 4 is eating
Philosopher 4 has finished eating
------------------------------------
Process exited after 12.78 seconds with return value 0
Press any key to continue . . .
```

```c
37          sem_wait(&chopstick[(phil+1)%5]);
38
39          eat(phil);
40          sleep(2);
41          printf("\nPhilosopher %d has finished eating",phil);
42
43          sem_post(&chopstick[(phil+1)%5]);
44          sem_post(&chopstick[phil]);
```