

Q Write a C program to simulate the concept of Dining-philosophers problem

```
#include <stdio.h>
#include <semaphore.h>
#include <pthread.h>
#define N 5
#define THINKING 2
#define HUNGRY 1
#define EATING 0
#define LEFT (phnum+4) % N
#define RIGHT (phnum+1) % N
```

```
int state[N];
int phil[N] = {0, 1, 2, 3, 4};
```

```
sem_t mutex;
sem_t s[N];
```

```
void test(int phnum)
{
```

```
    if (state[phnum] == HUNGRY
        && state[LEFT] != EATING
        && state[RIGHT] != EATING) {
        state[phnum] = EATING;
```

```
        sleep(2);
```

```
        printf("Philosopher %d takes fork %d and %d\n",
               phnum+1, LEFT+1, phnum+1);
        printf("Philosopher %d is Eating\n", phnum+1);
        sem_post(&s[phnum]);
```

void take_fork (int phnum)

```
{
    sem_wait (&mutex);
    state [phnum] = HUNGRY;
    printf ("philosopher %d is Hungry\n", phnum);
    test (phnum);
    sem_post (&mutex);
    sem_wait (&s[phnum]);
    sleep (1);
}
```

void put_fork (int phnum)

```
{
    sem_wait (&mutex);
    state [phnum] = THINKING;
    printf ("philosopher %d putting fork %d and %d\n", phnum+1, LEFT+1, phnum+1);
    printf ("philosopher %d is thinking\n", phnum+1);
```

test (LEFT);

test (RIGHT);

sem_post (&mutex);

}

void * philosopher (void * num)

```
{
    while (1) {
        int *i = num;
        sleep (1);
        take_fork (*i);
        sleep (1);
        put_fork (*i);
    }
}
```



```

int main()
{
    int i;
    pthread_t thread_id[N];
    sem_init (&mutex, 0, 1);
    for (i=0; i<N; i++)
        sem_init (&S[i], 0, 0);
    for (i=0; i<N; i++)
        pthread_create (&thread_id[i], NULL,
            philosopher, &phil[i]);
    printf("philosopher %d is thinking\n", i+1);
    for (i=0; i<N; i++)
        pthread_join (thread_id[i], NULL);
}

```

output

Philosopher 1 is thinking
 Philosopher 2 is thinking
 Philosopher 3 is thinking
 Philosopher 4 is thinking
 Philosopher 5 is thinking
 Philosopher 2 is Hungry
 Philosopher 1 is Hungry
 Philosopher 4 is Hungry
 Philosopher 5 is Hungry
 Philosopher 5 takes fork 4 and 5
 Philosopher 5 is eating
 Philosopher 3 is Hungry
 Philosopher 3 takes fork 2 and 3
 Philosopher 3 is Eating
 Philosopher 5 putting fork 4 and 5 down.
 Philosopher 5 is thinking.

10/10

2/8/23

Enter the number of philosophers: 5

Philosopher 1 is thinking

Philosopher 1 is eating

Philosopher 3 is thinking

Philosopher 3 is eating

Philosopher 5 is thinking

Philosopher 2 is thinking

Philosopher 4 is thinking

Philosopher 3 Finished eating

Philosopher 1 Finished eating

Philosopher 2 is eating

Philosopher 5 is eating

Philosopher 2 Finished eating

Philosopher 5 Finished eating

Philosopher 4 is eating