OS - DINING PHILOSOPHERS PROBLEM

4. Write a C program to simulate:
b) Dining-Philosophers problem using semaphores.

Code:

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

#define N 5
#define THINKING 0
#define HUNGRY 1
#define EATING 2

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

pthread_mutex_t mutex;
pthread_cond_t condition[N];

void test(int i){
    if(state[i]==HUNGRY && state[(i+4)%N]!=EATING && state[(i+1)%N]!=EATING){
        state[i]=EATING;
        pthread_cond_signal(&condition[i]);
    }
}

void takeforks(int i){
    pthread_mutex_lock(&mutex);
    state[i]=HUNGRY;
    printf("Philosopher %d is HUNGRY\n", i);
    test(i);
    while(state[i]!=EATING){
        pthread_cond_wait(&condition[i], &mutex);
    }
    printf("Philosopher %d takes fork and starts EATING\n", i);
    pthread_mutex_unlock(&mutex);
}

void putforks(int i){
```

```
    pthread_mutex_lock(&mutex);
    state[i]=THINKING;
    printf("Philosopher %d puts down forks and starts THINKING\n", i);
    test((i+4)%N);
    test((i+1)%N);
    pthread_mutex_unlock(&mutex);
}

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

    while(1){
        sleep(1);
        takeforks(i);
        sleep(2);
        putforks(i);
    }
}

int main(){
    int i;
    pthread_t thread_id[N];

    pthread_mutex_init(&mutex, NULL);
    for(int i=0;i<N;i++)
        pthread_cond_init(&condition[i], NULL);
    for(int i=0;i<N;i++)
        pthread_create(&thread_id[i], NULL, philosopher, &philosophers[i]);
    for(int i=0;i<N;i++)
        pthread_join(thread_id[i], NULL);
    pthread_mutex_destroy(&mutex);
    for(int i=0;i<N;i++)
        pthread_cond_destroy(&condition[i]);
    return 0;
}
```

Output:

```
Philosopher 2 is HUNGRY
Philosopher 2 takes fork and starts EATING
Philosopher 3 is HUNGRY
Philosopher 1 is HUNGRY
Philosopher 0 is HUNGRY
Philosopher 0 takes fork and starts EATING
Philosopher 4 is HUNGRY
Philosopher 2 puts down forks and starts THINKING
Philosopher 3 takes fork and starts EATING
Philosopher 0 puts down forks and starts THINKING
Philosopher 1 takes fork and starts EATING
Philosopher 2 is HUNGRY
Philosopher 0 is HUNGRY
Philosopher 3 puts down forks and starts THINKING
Philosopher 4 takes fork and starts EATING
Philosopher 1 puts down forks and starts THINKING
Philosopher 2 takes fork and starts EATING
Philosopher 3 is HUNGRY
Philosopher 1 is HUNGRY
Philosopher 4 puts down forks and starts THINKING
Philosopher 0 takes fork and starts EATING
Philosopher 2 puts down forks and starts THINKING
Philosopher 3 takes fork and starts EATING
Philosopher 4 is HUNGRY
Philosopher 2 is HUNGRY
Philosopher 0 puts down forks and starts THINKING
Philosopher 1 takes fork and starts EATING
Philosopher 3 puts down forks and starts THINKING
Philosopher 4 takes fork and starts EATING
Philosopher 0 is HUNGRY
```