Name: Aditya Somani
Roll No: T1851061
Div: A
PRN NO. 71901204L

ASSIGNMENT NO. 6

TITLE: Dining Philosophers Problem (Using Semaphore or mutex)

Program:

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

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

sem_t mutex;
sem_t S[N];

void * philosopher(void *num);
void take_fork(int);
void put_fork(int);
void test(int);

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

```
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_number[i]);
        printf("philosopher %d is thinking\n",i+1);
    }
    for(i = 0;i<N;i++)
    pthread_join(thread_id[i],NULL);

    return 0;
}


void *philosopher(void *num)
{
    while(1)
    {
        int *i = num;
        sleep(1);
```

```
        take_fork(*i);
        sleep(0);
        put_fork(*i);
    }
}

void take_fork(int ph_num)
{
    sem_wait(&mutex);
    state[ph_num] = HUNGRY;
    printf("Philosopher %d is hungry\n",ph_num+1);
    test(ph_num);
    sem_post(&mutex);
    sem_wait(&S[ph_num]);
    sleep(1);
}


void test(int ph_num)
{
    if (state[ph_num] == HUNGRY && state[(ph_num+4)%N]!= EATING && state)
    {
        state[ph_num] = EATING;
        sleep(2);
        printf("Philosopher %d takes fork %d and %d\n",ph_num+1,((ph_num)));
        printf("Philosopher %d is Eating\n",ph_num+1);
```

```
        sem_post(&S[ph_num]);
    }
}

void put_fork(int ph_num){
    sem_wait(&mutex);
    printf("Philosopher %d putting fork %d and %d
down\n",ph_num+1,((ph_num+4)%N)+1,ph_num+1);
    printf("Philosopher %d is thinking\n",ph_num+1);
    test((ph_num+4)%N);
    test((ph_num+1)%N);
    sem_post(&mutex);
}
```

Output :