**12. Design a C program to simulate the concept of Dining-Philosophers**

**Problem**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <unistd.h>

#define NUM\_PHILOSOPHERS 5pthread\_mutex\_t chopsticks[NUM\_PHILOSOPHERS];

void\* philosopherLifeCycle(void\* arg) {int id =

\*((int\*)arg);

int left\_chopstick = id;

int right\_chopstick = (id + 1) % NUM\_PHILOSOPHERS;

while (1) {

// Think

printf("Philosopher %d is thinking...\n", id);

// Pick up chopsticks

pthread\_mutex\_lock(&chopsticks[left\_chopstick]);

pthread\_mutex\_lock(&chopsticks[right\_chopstick]);

// Eat

printf("Philosopher %d is eating...\n", id);

sleep(rand() % 3 + 1); // Eating time

// Put down chopsticks

pthread\_mutex\_unlock(&chopsticks[left\_chopstick]);

pthread\_mutex\_unlock(&chopsticks[right\_chopstick]);

// Repeat the cycle

}

}

int main() {

pthread\_t philosophers[NUM\_PHILOSOPHERS];int

philosopher\_ids[NUM\_PHILOSOPHERS];

for (int i = 0; i < NUM\_PHILOSOPHERS; ++i) {

pthread\_mutex\_init(&chopsticks[i], NULL);

}

for (int i = 0; i < NUM\_PHILOSOPHERS; ++i) {

philosopher\_ids[i] = i;pthread\_create(&philosophers[i], NULL, philosopherLifeCycle,(void\*)&philosopher\_ids[i]);

}

for (int i = 0; i < NUM\_PHILOSOPHERS; ++i) {

pthread\_join(philosophers[i], NULL);

}

for (int i = 0; i < NUM\_PHILOSOPHERS; ++i) {

pthread\_mutex\_destroy(&chopsticks[i]);

}

return 0;

}

