**4. Write a C program to simulate: a) Producer-Consumer problem using semaphores.  
b) Dining-Philosopher’s problem**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <pthread.h>**

**#include <semaphore.h>**

**#include <unistd.h>**

**int buffer;**

**sem\_t empty, full;**

**pthread\_mutex\_t mutex;**

**void\* producer(void\* arg) {**

**int item = rand() % 100;**

**sem\_wait(&empty);**

**pthread\_mutex\_lock(&mutex);**

**buffer = item;**

**printf("Producer produced: %d\n", item);**

**pthread\_mutex\_unlock(&mutex);**

**sem\_post(&full);**

**return NULL;**

**}**

**void\* consumer(void\* arg) {**

**int item;**

**sem\_wait(&full);**

**pthread\_mutex\_lock(&mutex);**

**item = buffer;**

**printf("Consumer consumed: %d\n", item);**

**pthread\_mutex\_unlock(&mutex);**

**sem\_post(&empty);**

**return NULL;**

**}**

**void run\_producer\_consumer() {**

**pthread\_t p, c;**

**sem\_init(&empty, 0, 1);**

**sem\_init(&full, 0, 0);**

**pthread\_mutex\_init(&mutex, NULL);**

**pthread\_create(&p, NULL, producer, NULL);**

**pthread\_create(&c, NULL, consumer, NULL);**

**pthread\_join(p, NULL);**

**pthread\_join(c, NULL);**

**sem\_destroy(&empty);**

**sem\_destroy(&full);**

**pthread\_mutex\_destroy(&mutex);**

**}**

**sem\_t forks[2];**

**void\* philosopher(void\* arg) {**

**int id = \*(int\*)arg;**

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

**sleep(1);**

**sem\_wait(&forks[0]);**

**sem\_wait(&forks[1]);**

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

**sleep(1);**

**sem\_post(&forks[0]);**

**sem\_post(&forks[1]);**

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

**return NULL;**

**}**

**void run\_dining\_philosophers() {**

**pthread\_t phil;**

**int id = 1;**

**sem\_init(&forks[0], 0, 1);**

**sem\_init(&forks[1], 0, 1);**

**pthread\_create(&phil, NULL, philosopher, &id);**

**pthread\_join(phil, NULL);**

**sem\_destroy(&forks[0]);**

**sem\_destroy(&forks[1]);**

**}**

**int main() {**

**int choice;**

**while (1) {**

**printf("\n1. Producer-Consumer\n2. Dining Philosopher\n3. Exit\nChoice: ");**

**scanf("%d", &choice);**

**switch (choice) {**

**case 1: run\_producer\_consumer(); break;**

**case 2: run\_dining\_philosophers(); break;**

**case 3: exit(0);**

**default: printf("Invalid choice!\n");**

**}**

**}**

**}**

