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>


// Producer-Consumer Problem (Simple)

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+1);

   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+1);

   pthread_mutex_unlock(&mutex);
   sem_post(&empty);
   return NULL;
}
```

```
void run_producer_consumer() {
    pthread_t prod, cons;
    sem_init(&empty, 0, 1);
    sem_init(&full, 0, 0);
    pthread_mutex_init(&mutex, NULL);

    pthread_create(&prod, NULL, producer, NULL);
    pthread_create(&cons, NULL, consumer, NULL);

    pthread_join(prod, NULL);
    pthread_join(cons, NULL);

    sem_destroy(&empty);
    sem_destroy(&full);
    pthread_mutex_destroy(&mutex);
}


// Dining Philosophers Problem (Simple)

sem_t fork1, fork2;

void* philosopher(void* arg) {
    int id = *(int*)arg;
    printf("Philosopher %d is thinking\n", id);
    sleep(1);

    sem_wait(&fork1);
    sem_wait(&fork2);

    printf("Philosopher %d is eating\n", id);
    sleep(1);

    sem_post(&fork1);
    sem_post(&fork2);

    printf("Philosopher %d finished eating\n", id);
    return NULL;
}

void run_dining_philosophers() {
```

```
    pthread_t phil;
    int id = 1;
    sem_init(&fork1, 0, 1);
    sem_init(&fork2, 0, 1);

    pthread_create(&phil, NULL, philosopher, &id);
    pthread_join(phil, NULL);

    sem_destroy(&fork1);
    sem_destroy(&fork2);
}

int main() {
    int choice;
    while (1) {
        printf("\n1. Producer-Consumer Problem\n");
        printf("2. Dining Philosophers Problem\n");
        printf("3. Exit\n");
        printf("Enter your choice: ");
        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");
        }
    }
    return 0;
}
```

Output:

```
PS C:\Users\Admin\Desktop\OS LAB> ./a.exe

  1. Producer-Consumer Problem
  2. Dining Philosophers Problem
  3. Exit
  Enter your choice: 1
  Producer produced: 42
  Consumer consumed: 42

  1. Producer-Consumer Problem
  2. Dining Philosophers Problem
  3. Exit
  Enter your choice: 2
  Philosopher 1 is thinking
  Philosopher 1 is eating
  Philosopher 1 finished eating

  1. Producer-Consumer Problem
  2. Dining Philosophers Problem
  3. Exit
  Enter your choice: 3
PS C:\Users\Admin\Desktop\OS LAB>
```