3. To Write a C program to simulate producer-consumer problem using semaphores.

PROGRAM

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


#define BUFFER_SIZE 10


int buffer[BUFFER_SIZE];
int in = 0;
int out = 0;


sem_t empty;
sem_t full;
sem_t mutex;


void* producer(void* arg) {
    while (1) {
        sleep(rand() % 3);
        int item = rand() % 100;
        sem_wait(&empty);  // sem_wait decrements (locks) a
semaphore
```

```
        sem_wait(&mutex);
        buffer[in] = item;
        in = (in + 1) % BUFFER_SIZE;
        printf("Produced: %d\n", item);

// sem_post increments (unlocks) a semaphore, signaling that a resource is available
        sem_post(&mutex);
        sem_post(&full);
    }
    return NULL;
}

void* consumer(void* arg) {
    while (1) {
        sleep(rand() % 3);
        sem_wait(&full);
        sem_wait(&mutex);
        int item = buffer[out];
        out = (out + 1) % BUFFER_SIZE;
        printf("Consumed: %d\n", item);
        sem_post(&mutex);
        sem_post(&empty);
    }
    return NULL;
}
```

```
int main() {
    pthread_t producer_thread, consumer_thread;

    sem_init(&empty, 0, BUFFER_SIZE);
    sem_init(&full, 0, 0);
    sem_init(&mutex, 0, 1);

    pthread_create(&producer_thread, NULL, producer, NULL);
    pthread_create(&consumer_thread, NULL, consumer, NULL);

    pthread_join(producer_thread, NULL);
    pthread_join(consumer_thread, NULL);

    sem_destroy(&empty);
    sem_destroy(&full);
    sem_destroy(&mutex);
        getch();
    return 0;
}
```

VIVA Questions:

1. What is race condition? How do semaphores prevent it.
2. what is the difference between sem_post and sem_wait

semaphore operations?