

**COURSE: OPERATING SYSTEMS**  
**LAB: 05**  
**SUBMITTED BY: EBAAD KHAN**  
**ROLL: DT-22045**

1) Implement the above code and paste the screen shot of the output.

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

sem_t x, y;
int readercount = 0;

void *reader(void* param) {
    sem_wait(&x);
    readercount++;

    if (readercount == 1)
        sem_wait(&y); // First reader locks the resource

    sem_post(&x);

    printf("%d reader(s) are inside\n", readercount);
    usleep(3000000); // Simulating reading time

    sem_wait(&x);
    readercount--;

    if (readercount == 0) {
        sem_post(&y); // Last reader releases the lock
    }

    sem_post(&x);
    printf("%d reader(s) are leaving\n", readercount + 1);
    return NULL;
}

void *writer(void* param) {
    printf("Writer is trying to enter\n");
    sem_wait(&y); // Writer locks the resource
    printf("Writer has entered\n");

    usleep(3000000); // Simulating writing time

    sem_post(&y); // Writer releases the resource
    printf("Writer is leaving\n");
    return NULL;
}
```

```

int main() {
    int num_readers, num_writers, i;

    printf("Enter the number of readers: ");
    scanf("%d", &num_readers);

    printf("Enter the number of writers: ");
    scanf("%d", &num_writers);

    pthread_t reader_threads[num_readers], writer_threads[num_writers];

    // Initialize semaphores
    sem_init(&x, 0, 1);
    sem_init(&y, 0, 1);

    // Creating reader threads
    for (i = 0; i < num_readers; i++) {
        pthread_create(&reader_threads[i], NULL, reader, NULL);
    }

    // Creating writer threads
    for (i = 0; i < num_writers; i++) {
        pthread_create(&writer_threads[i], NULL, writer, NULL);
    }

    // Joining reader threads
    for (i = 0; i < num_readers; i++) {
        pthread_join(reader_threads[i], NULL);
    }

    // Joining writer threads
    for (i = 0; i < num_writers; i++) {
        pthread_join(writer_threads[i], NULL);
    }

    // Destroy semaphores
    sem_destroy(&x);
    sem_destroy(&y);

    return 0;
}

```

#### Output

```

Enter the number of readers: 3
Enter the number of writers: 2
1 reader(s) are inside
2 reader(s) are inside
3 reader(s) are inside
Writer is trying to enter
Writer is trying to enter
3 reader(s) are leaving
2 reader(s) are leaving
1 reader(s) are leaving
Writer has entered
Writer is leaving
Writer has entered
Writer is leaving

```

```

=== Code Execution Successful ===

```