NAME : TASKEEN SADIQ **ROLL NO : DT - 22004**

OS LAB : 05

QUESTION : 01

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

sem_t x, y;
pthread_t writerthreads[100], readerthreads[100];
int readercount = 0;
void *reader(void *param)
{
    sem_wait(&x);
    readercount++;

    if (readercount == 1)
        sem_wait(&y);

    sem_post(&x);
    printf("%d reader is inside\n", readercount);
    usleep(3);
```

```
    sem_wait(&x);
    readercount--;

    if (readercount == 0) {
       sem_post(&y);
}
    sem_post(&x);

    printf("%d Reader is leaving\n", readercount + 1);
    return NULL;
}

void *writer(void *param)
{
    printf("Writer is trying to enter\n");
    sem_wait(&y);
    printf("Writer has entered\n");
    sem_post(&y);
    printf("Writer is leaving\n");
    return NULL;
}

int main()
{
```

```
    int n2, i;

    printf("Enter the number of readers: ");
    scanf("%d", &n2);

    sem_init(&x, 0, 1);  // Semaphore for reader count
    sem_init(&y, 0, 1);  // Semaphore for controlling writer access
    for (i = 0; i < n2; i++)
    {
        pthread_create(&readerthreads[i], NULL, reader, NULL);
        pthread_create(&writerthreads[i], NULL, writer, NULL);
    }
    for (i = 0; i < n2; i++)
    {
        pthread_join(readerthreads[i], NULL);
        pthread_join(writerthreads[i], NULL);
    }

    sem_destroy(&x);
    sem_destroy(&y);

    return 0;
}
```

NAME : TASKEEN SADIQ **ROLL NO : DT - 22004**

```
Enter the number of readers: 4
1 reader is inside
2 Reader is leaving
2 reader is inside
2 Reader is leaving
Writer is trying to enter
Writer is trying to enter
2 reader is inside
2 Reader is leaving
Writer is trying to enter
2 reader is inside
1 Reader is leaving
Writer has entered
Writer is leaving
Writer is trying to enter
Writer has entered
Writer is leaving
Writer has entered
Writer is leaving
Writer has entered
Writer is leaving

--------------------------------
Process exited after 2.304 seconds with return value 0
Press any key to continue . . .
```