LAB-05
NAME : MUHMMAD HUZAIFA SALMAN
ROLL NO :DT-34

Exercise:

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

PROGRAM:

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

sem_t x, y;
pthread_t readerthreads[100], writerthreads[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(3000); // sleep for 3000 microseconds (adjust as needed)

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

    sem_init(&x, 0, 1);
    sem_init(&y, 0, 1);

    // Create n reader threads and n writer threads.
    for (i = 0; i < n; i++)
    {
        pthread_create(&readerthreads[i], NULL, reader, NULL);
        pthread_create(&writerthreads[i], NULL, writer, NULL);
    }

    // Wait for all threads to finish.
    for (i = 0; i < n; i++)
    {
        pthread_join(readerthreads[i], NULL);
        pthread_join(writerthreads[i], NULL);
    }

    return 0;
}
```

OUTPUT:

```
Enter the number of readers: 4

Writer is trying to enter
Writer has entered
Writer is leaving
1 reader is inside
Writer is trying to enter
2 reader is inside
Writer is trying to enter
3 reader is inside
Writer is trying to enter
4 reader is inside
4 Reader is leaving
3 Reader is leaving
2 Reader is leaving
1 Reader is leaving
Writer has entered
Writer is leaving
Writer has entered
```