READER-WRITER

```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
sem_t wrt;
pthread_mutex_t mutex;
int read_count = 0;
int shared_data = 0;
void* reader(void* arg) {
    int id = *((int*)arg);
    pthread_mutex_lock(&mutex);
    read_count++;
    if (read_count == 1)
        sem_wait(&wrt); // first reader locks writer
    pthread_mutex_unlock(&mutex);
    printf("Reader %d reads data = %d\n", id, shared_data);
    sleep(1);
    pthread_mutex_lock(&mutex);
    read_count--;
    if (read_count == 0)
        sem_post(&wrt);
    pthread_mutex_unlock(&mutex);

    return NULL;
}
void* writer(void* arg) {
    int id = *((int*)arg);
    sem_wait(&wrt);
    shared_data += 10;
    printf("Writer %d writes data = %d\n", id, shared_data);
    sleep(1);
    sem_post(&wrt);
    return NULL;
}
int main() {
    pthread_t rtid[5], wtid[5];
    int ids[5];
    sem_init(&wrt, 0, 1);
    pthread_mutex_init(&mutex, NULL);
    for (int i = 0; i < 5; i++) {
        ids[i] = i + 1;
        pthread_create(&wtid[i], NULL, writer, &ids[i]);
        pthread_create(&rtid[i], NULL, reader, &ids[i]);
    }
    for (int i = 0; i < 5; i++) {
        pthread_join(wtid[i], NULL);
        pthread_join(rtid[i], NULL);
    }
    sem_destroy(&wrt);
    pthread_mutex_destroy(&mutex);
    return 0;
}
```