1. Construct a C program to simulate Reader-Writer problem using Semaphores.

#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>

int readCount = 0;

sem\_t mutex, wrt;

int data = 0;

void\* reader(void\* arg) {

int id = *((int*)arg);

sem\_wait(&mutex);

readCount++;

if (readCount == 1)

sem\_wait(&wrt);

sem\_post(&mutex);

printf("Reader %d: read data = %d\n", id, data);

sleep(1)

sem\_wait(&mutex);

readCount--;

if (readCount == 0)

sem\_post(&wrt); // Last reader releases writer

sem\_post(&mutex);

return NULL;

}

void\* writer(void\* arg) {

int id = *((int*)arg);

sem\_wait(&wrt);

data += 5;

printf("Writer %d: wrote data = %d\n", id, data);

sleep(1);

sem\_post(&wrt);

return NULL:

}

int main() {

pthread\_t r[5], w[2];

int r\_id[5] = {1, 2, 3, 4, 5};

int w\_id[2] = {1, 2};

sem\_init(&mutex, 0, 1);

sem\_init(&wrt, 0, 1);

for (int i = 0; i < 2; i++)

pthread\_create(&w[i], NULL, writer, &w\_id[i]);

for (int i = 0; i < 5; i++)

pthread\_create(&r[i], NULL, reader, &r\_id[i]);

for (int i = 0; i < 2; i++)  
 pthread\_join(w[i], NULL);  
 for (int i = 0; i < 5; i++)  
 pthread\_join(r[i], NULL);  
  
 sem\_destroy(&mutex);  
 sem\_destroy(&wrt);  
 return 0;

}