19. Write a C program to create two threads to access shared memory which is an integer in a synchronized fashion using semaphore. In the first thread print the doubled the integer data after reading from the shared memory. In the second thread, print the five times of the integer data after reading from the shared memory

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

int shared\_data = 10; // the shared integer variable

sem\_t semaphore; // the semaphore variable

// the first thread function

void\* double\_thread(void\* arg) {

int\* data = (int\*)arg;

sem\_wait(&semaphore); // acquire the semaphore

shared\_data = \*data \* 2; // double the shared data

printf("Double Thread: Shared data is now %d\n", shared\_data);

sem\_post(&semaphore); // release the semaphore

return NULL;

}

// the second thread function

void\* five\_thread(void\* arg) {

int\* data = (int\*)arg;

sem\_wait(&semaphore); // acquire the semaphore

shared\_data = \*data \* 5; // multiply the shared data by 5

printf("Five Thread: Shared data is now %d\n", shared\_data);

sem\_post(&semaphore); // release the semaphore

return NULL;

}

int main() {

pthread\_t tid1, tid2; // thread ids

sem\_init(&semaphore, 0, 1); // initialize the semaphore

// create the first thread

int data1 = shared\_data;

pthread\_create(&tid1, NULL, double\_thread, &data1);

// create the second thread

int data2 = shared\_data;

pthread\_create(&tid2, NULL, five\_thread, &data2);

// wait for the threads to finish

pthread\_join(tid1, NULL);

pthread\_join(tid2, NULL);

sem\_destroy(&semaphore); // destroy the semaphore

return 0;

}