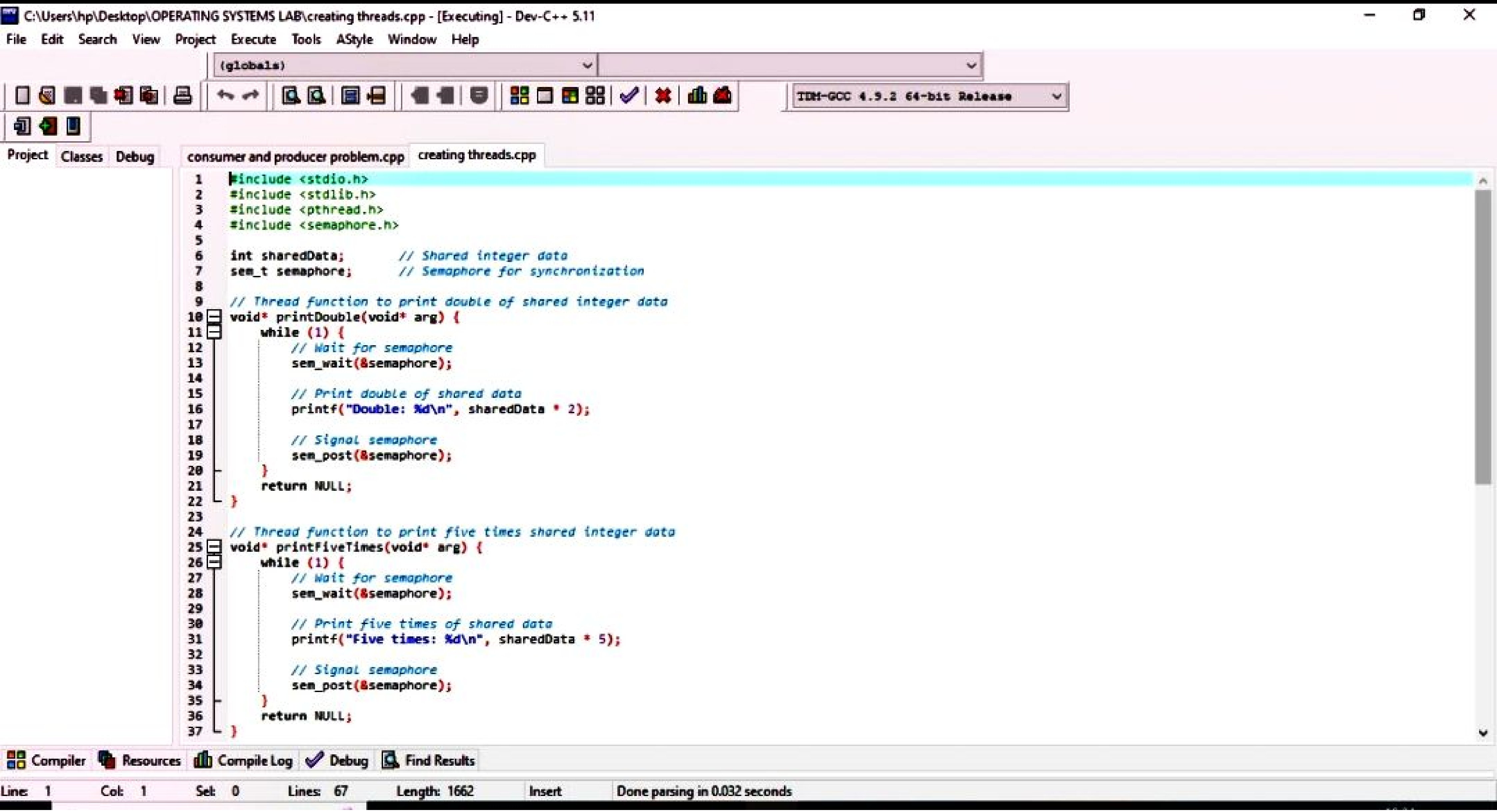
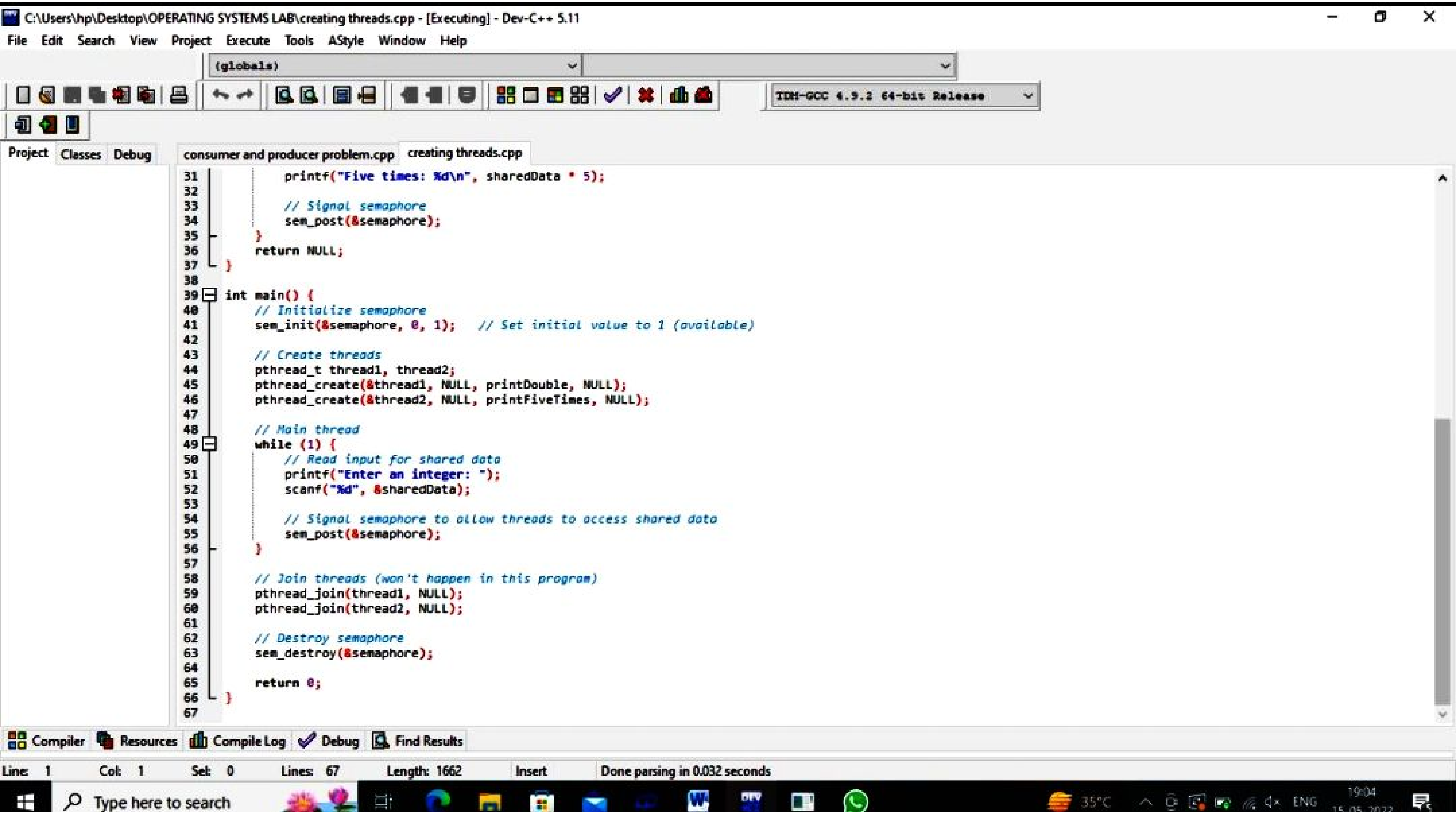
19. Write a C program to create two threads to access shared memory which is an integer in a synchronized fashion using semaphore. In the first thread print the doubled the integer data after reading from the shared memory. In the second thread, print the five times of the integer data after reading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


C:\Users\hp\Desktop\OPERATING SYSTEMS LAB\creating threads.cpp - [Executing] - Dev-C++ 5.11
File
Edit
Search
View
Project
Execute
Tools
AStyle
Window
Help
(globals)
TDM-GCC 4.9.2 64-bit Release
Project
Classes
Debug
consumer and producer problem.cpp
creating threads.cpp
Compiler
Resources
Compile Log
Debug
Find Results

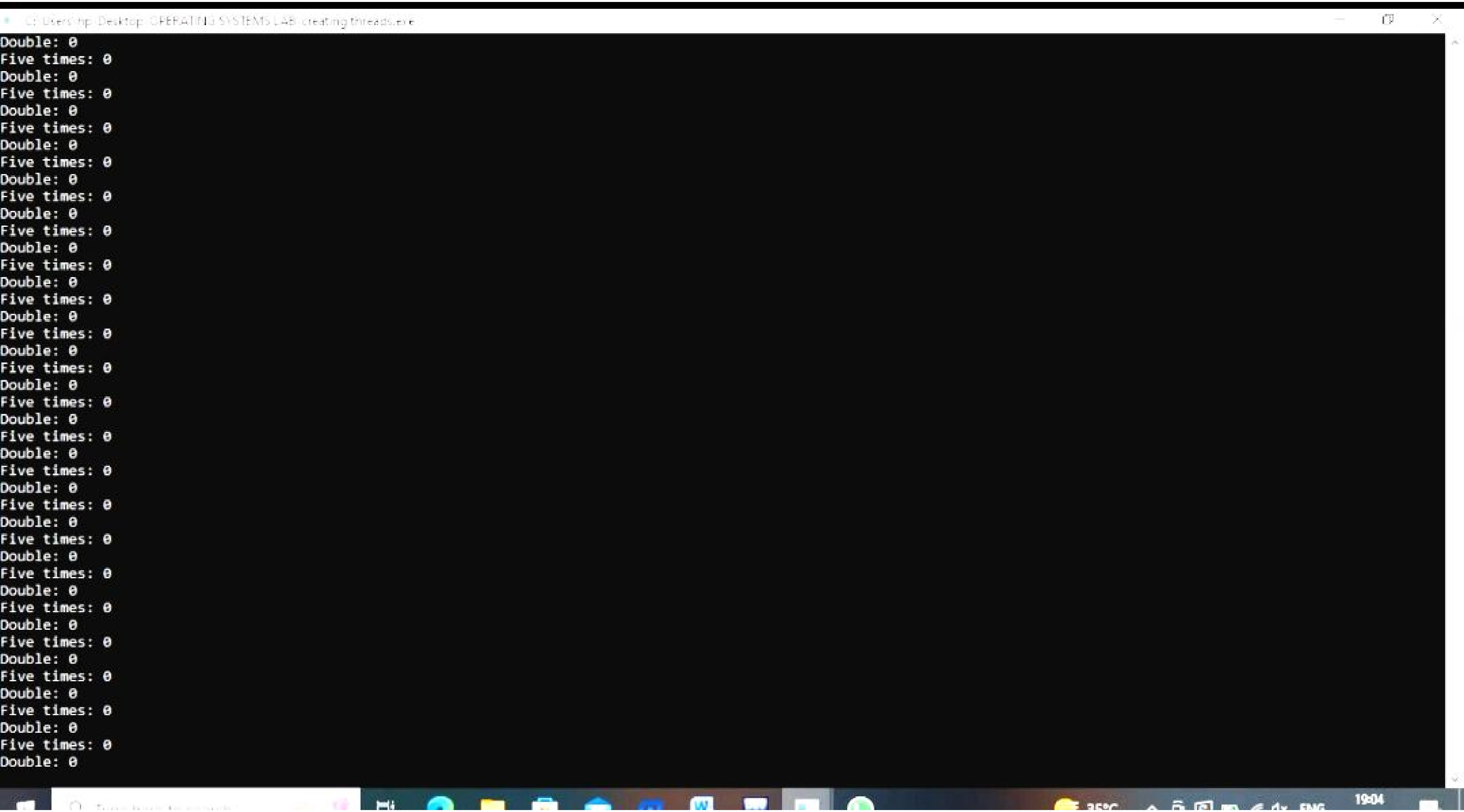
```c
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>

int sharedData;       // Shared integer data
sem_t semaphore;      // Semaphore for synchronization

// Thread function to print double of shared integer data
void* printDouble(void* arg) {
    while (1) {
        // Wait for semaphore
        sem_wait(&semaphore);

        // Print double of shared data
        printf("Double: %d\n", sharedData * 2);

        // Signal semaphore
        sem_post(&semaphore);
    }
    return NULL;
}

// Thread function to print five times shared integer data
void* printFiveTimes(void* arg) {
    while (1) {
        // Wait for semaphore
        sem_wait(&semaphore);

        // Print five times of shared data
        printf("Five times: %d\n", sharedData * 5);

        // Signal semaphore
        sem_post(&semaphore);
    }
    return NULL;
}
```

Line: 1 Col: 1 Sel: 0 Lines: 67 Length: 1662 Insert Done parsing in 0.032 seconds


C:\Users\hp\Desktop\OPERATING SYSTEMS LAB\creating threads.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
TDM-GCC 4.9.2 64-bit Release
Project Classes Debug
consumer and producer problem.cpp
creating threads.cpp
Compiler Resources Compile Log Debug Find Results
Type here to search
35°C
ENG
19:04

```
        printf("Five times: %d\n", sharedData * 5);

        // Signal semaphore
        sem_post(&semaphore);
    }
    return NULL;
}

int main() {
    // Initialize semaphore
    sem_init(&semaphore, 0, 1);   // Set initial value to 1 (available)

    // Create threads
    pthread_t thread1, thread2;
    pthread_create(&thread1, NULL, printDouble, NULL);
    pthread_create(&thread2, NULL, printFiveTimes, NULL);

    // Main thread
    while (1) {
        // Read input for shared data
        printf("Enter an integer: ");
        scanf("%d", &sharedData);

        // Signal semaphore to allow threads to access shared data
        sem_post(&semaphore);
    }

    // Join threads (won't happen in this program)
    pthread_join(thread1, NULL);
    pthread_join(thread2, NULL);

    // Destroy semaphore
    sem_destroy(&semaphore);

    return 0;
}

```

Line: 1 Col: 1 Sel: 0 Lines: 67 Length: 1662 Insert Done parsing in 0.032 seconds


C:\Users\hp\Desktop\OPERATING SYSTEMS LAB\creating threads.exe
19:04

```
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
Five times: 0
Double: 0
```