```cpp
// Example code uses CRITICAL SECTION & MUTUAL EXCLUSION.
// programmer: supakit@it.kmitl.ac.th
// 2017 Mar 30  completed

#include <stdio.h>
#include <thread>
#include <mutex>


#define NUM_THREADS 5

static int threadData[NUM_THREADS];
int g;
std::mutex  g_mutex;


void func_mycounter(int id) {
    std::unique_lock<std::mutex> chkCritical(g_mutex, std::defer_lock);
    int i;
    printf("\n thread[%d]--start", id);
    chkCritical.lock();
    printf("\n [%d] running", id);
    for (int i=0;i < 1e8;i++){
        g = g + 1 ;
    }
    chkCritical.unlock();
    printf("\n\t\t\t thread[%d]--finish", id);
}

int main(void) {
    std::thread threadId[NUM_THREADS];
    printf("\n-------------MULTITHREAD--------------------");
    g = 0;
    printf("\n start g = %d", g);

    for (int i = 0; i < NUM_THREADS; i++) {
        threadId[i] = std::thread(func_mycounter, i);
    }

    for (int i = 0; i < NUM_THREADS; i++){
        threadId[i].join();
    }

    printf("\n finish g = %d", g);

    /*
    printf("\n-------------SINGLE THREAD--------------------");
    g = 0;
    printf("\n start g = %d", g);
    for (int i = 0; i < 5; i++) {
        func_mycounter(i);
    }
    printf("\n finish g = %d", g);
    */
    getchar();
    return 0;
```

```
57  }
```