19.Design a C program to implement process synchronization using mutex locks.

#include <stdio.h>

#include <pthread.h>

pthread\_mutex\_t lock;

void\* critical\_section(void\* arg) {

pthread\_mutex\_lock(&lock);

printf("Thread %d in critical section\n", \*(int\*)arg);

pthread\_mutex\_unlock(&lock);

return NULL;

}

int main() {

pthread\_t threads[5];

int thread\_ids[5];

pthread\_mutex\_init(&lock, NULL);

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

thread\_ids[i] = i;

pthread\_create(&threads[i], NULL, critical\_section, &thread\_ids[i]);

}

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

pthread\_join(threads[i], NULL);

}

pthread\_mutex\_destroy(&lock);

return 0;

}