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

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#define NUM\_THREADS 5

#define INCREMENTS 1000

int counter = 0;

pthread\_mutex\_t lock;

void\* incrementCounter(void\* arg) {

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

pthread\_mutex\_lock(&lock);

counter++;

pthread\_mutex\_unlock(&lock);

}

return NULL;

}

int main() {

pthread\_t threads[NUM\_THREADS];

pthread\_mutex\_init(&lock, NULL);  
  
 for (int i = 0; i < NUM\_THREADS; i++) {  
 pthread\_create(&threads[i], NULL, incrementCounter, NULL);  
}  
  
for (int i = 0; i < NUM\_THREADS; i++) {  
 pthread\_join(threads[i], NULL);  
}  
  
pthread\_mutex\_destroy(&lock);  
  
printf("Final Counter Value: %d\n", counter);  
return 0;

}