```c
// ***********************************************************************
// Example of use of pthreads library.
// ***********************************************************************

#include <pthread.h>
#include <stdio.h>
#include <semaphore.h>
#include <stdint.h>

#define Max 10
#define Limit 1000000

int counter = 0;
sem_t mutex;
pthread_t tid[Max];

// Function that each thread will execute.
void *increment (void *arg)
{ int j, id = (intptr_t)arg;

  printf ("Thread %d  started!\n", id);
  for (j=0; j<Limit; j++)
  { sem_wait( &mutex);
    counter++;
    sem_post( &mutex );
  }
  printf ("Thread %d  is ending!\n", id);
}

void main()
{ int i, numT;

  printf ("Enter number of threads to create: ");
  scanf ("%d", &numT);
  if (numT <= 0 || numT > Max) numT = 5;  // if numT is invalid, use default

  // Initialize the semaphore
  // 2nd parameter: whether the semaphore is shared with child processes
  // 3rd parameter: the initial value of the semaphore
  sem_init( &mutex, 0, 1 );

  // Create num-threads of identical threads using pthread_create
  // 1st parameter: returned thread id
  // 2nd parameter: configure thread attributes, NULL for default values
  // 3rd parameter: routine for the thread to execute
  // 4th parameter: argument for the thread routine, has to be
  //   of type (void *), can be NULL if no argument to be passed
  for ( i = 0; i < numT; i++ ){
    pthread_create (&tid[i], NULL, increment, (void *)(intptr_t) i);
  }

  // Wait for all the threads to exit using pthread_join
  // 2nd parameter: to obtain the return value from the thread (void **)
  for ( i = 0; i < numT; i++ )
    pthread_join(tid[i], NULL);

  // Display final contents of counter
  printf ("\nFinal value of counter: %d (should be %d)\n",
          counter, numT * Limit);
}
```