```
#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <string.h>

int primes_count;

void *primesThreadFunction(void *args){

  int n = *((int *) args);

  int flag = 0;
  int i;

  for(i=2; i<=n/2; ++i){
    if(n%i==0){
      flag=1;
      break;
    }
  }

  if (flag==0){
    primes_count ++;
    printf("%d is a prime number.",n);
  }
  else
  printf("%d is not a prime number.",n);

  pthread_exit(NULL);
}

int main(int argc, char *argv[]){

  int n_threads = atoi(argv[0]);
  pthread_t prime_threads[n_threads];

  int t;
  int returnCode;

  //Create n threads and let them calculate primes
  for(t=1;t<n_threads;t++){

    returnCode = pthread_create(&prime_threads[t], NULL,
      primesThreadFunction, (void *)t);
      if (returnCode) {
        printf("ERROR return code from pthread_create() : %d\n",returnCode);
        exit(-1);
      }
    }

    //Wait for all threads to exit
    for(t=0;t<=n_threads; t++)
    pthread_join(prime_threads[t], NULL);

    printf("Successfully exited all threads!\n");
    printf("Number of primes between 1 and %d = %d\n", n_threads, primes_count);
    return(0);
  }
```