University of Wolverhampton

School of Mathematics and Computer Science

Student Id:2052261
Name: Mahan Timalsena

6CS005 High Performance Computing Week 2 Workshop

Revision on Multithreading

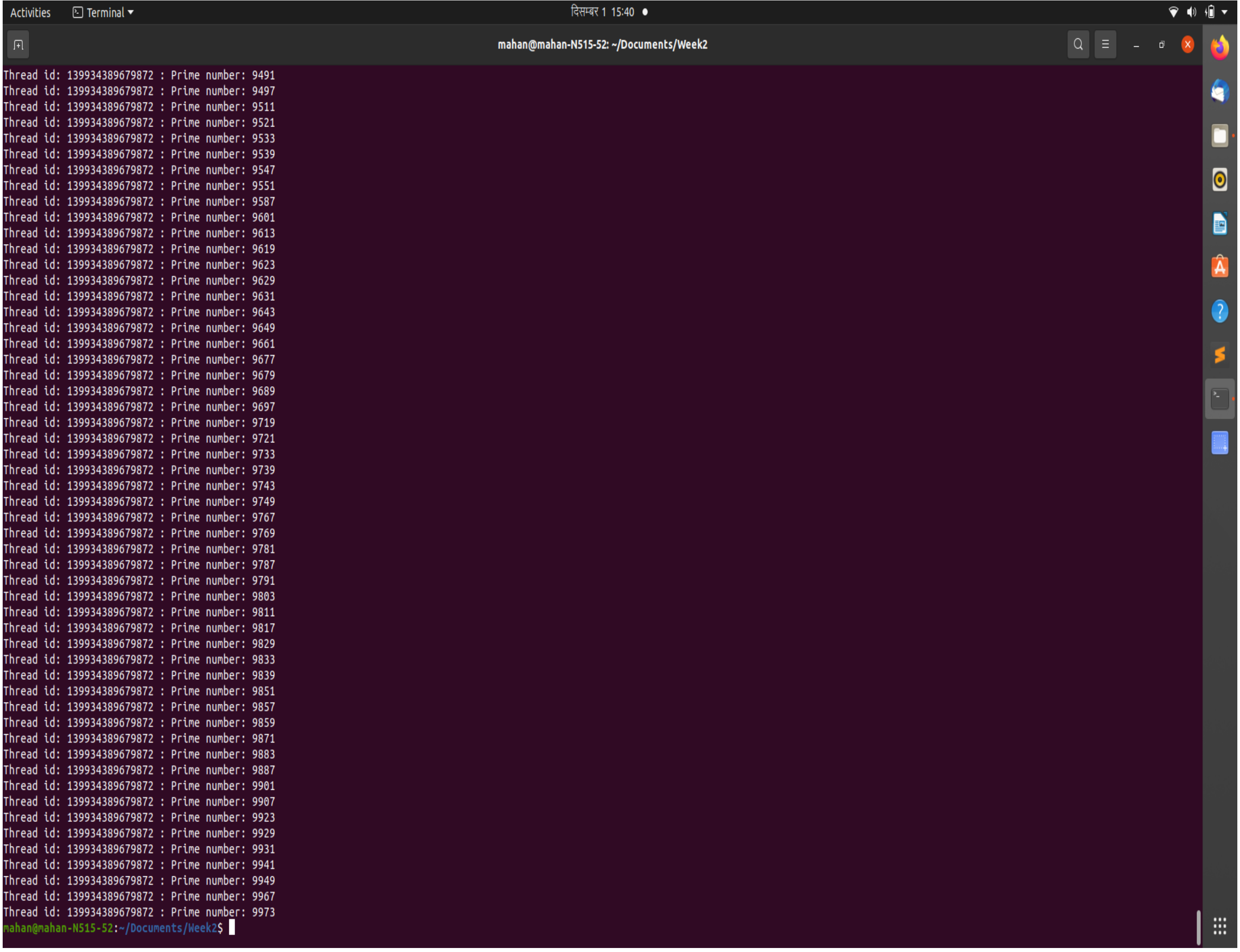
Tasks – Multithreading

1) Write a multithreaded C program to print out all the prime numbers between 1 to 10000. Use exactly 3 threads.

Code:

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

struct threadArgs {
int start;
int finish;
};

void *threadMain(void *p){
struct threadArgs *pargs = p;
int i, c;
int nstart=pargs->start, nfinish=pargs->finish;
pthread_t tid = pthread_self();

for(i=nstart; i<=nfinish; i++){
    for(c=2; c<=i-1; c++) {
    if ( i%c==0 )
    break;
    }
    if ( c==i )
    printf("Thread id: %ld : Prime number: %d\n",tid, i);
    }
return 0;
}

void main(int argc, char **argv){
int numThreads = 3;
int i;
pthread_t thrID[numThreads];
struct threadArgs targs[100];
if (numThreads > 0 && numThreads <= 100){
int chunksize = 10000/numThreads ;
for (i=0; i < numThreads; i++){
targs[i].start = i * chunksize;
targs[i].finish = (i * chunksize) + chunksize;
pthread_create(&thrID[i], NULL, threadMain, &targs[i]);
}
for (i=0; i < numThreads; i++){
pthread_join(thrID[i], NULL);
}
}
}
```

Output:



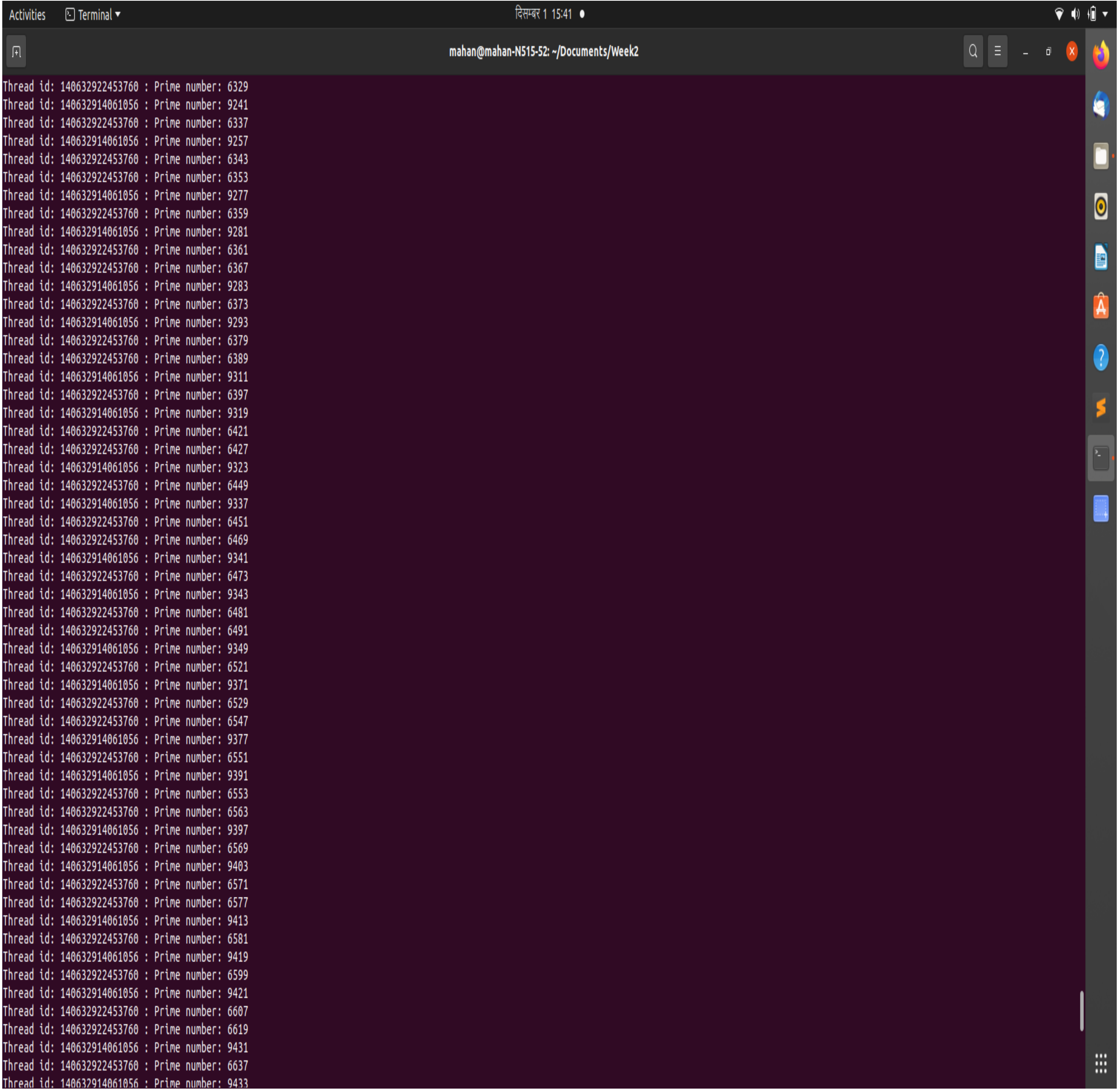
```
Thread id: 139934389679872 : Prime number: 9491
Thread id: 139934389679872 : Prime number: 9497
Thread id: 139934389679872 : Prime number: 9511
Thread id: 139934389679872 : Prime number: 9521
Thread id: 139934389679872 : Prime number: 9533
Thread id: 139934389679872 : Prime number: 9539
Thread id: 139934389679872 : Prime number: 9547
Thread id: 139934389679872 : Prime number: 9551
Thread id: 139934389679872 : Prime number: 9587
Thread id: 139934389679872 : Prime number: 9601
Thread id: 139934389679872 : Prime number: 9613
Thread id: 139934389679872 : Prime number: 9619
Thread id: 139934389679872 : Prime number: 9623
Thread id: 139934389679872 : Prime number: 9629
Thread id: 139934389679872 : Prime number: 9631
Thread id: 139934389679872 : Prime number: 9643
Thread id: 139934389679872 : Prime number: 9649
Thread id: 139934389679872 : Prime number: 9661
Thread id: 139934389679872 : Prime number: 9677
Thread id: 139934389679872 : Prime number: 9679
Thread id: 139934389679872 : Prime number: 9689
Thread id: 139934389679872 : Prime number: 9697
Thread id: 139934389679872 : Prime number: 9719
Thread id: 139934389679872 : Prime number: 9721
Thread id: 139934389679872 : Prime number: 9733
Thread id: 139934389679872 : Prime number: 9739
Thread id: 139934389679872 : Prime number: 9743
Thread id: 139934389679872 : Prime number: 9749
Thread id: 139934389679872 : Prime number: 9767
Thread id: 139934389679872 : Prime number: 9769
Thread id: 139934389679872 : Prime number: 9781
Thread id: 139934389679872 : Prime number: 9787
Thread id: 139934389679872 : Prime number: 9791
Thread id: 139934389679872 : Prime number: 9803
Thread id: 139934389679872 : Prime number: 9811
Thread id: 139934389679872 : Prime number: 9817
Thread id: 139934389679872 : Prime number: 9829
Thread id: 139934389679872 : Prime number: 9833
Thread id: 139934389679872 : Prime number: 9839
Thread id: 139934389679872 : Prime number: 9851
Thread id: 139934389679872 : Prime number: 9857
Thread id: 139934389679872 : Prime number: 9859
Thread id: 139934389679872 : Prime number: 9871
Thread id: 139934389679872 : Prime number: 9883
Thread id: 139934389679872 : Prime number: 9887
Thread id: 139934389679872 : Prime number: 9901
Thread id: 139934389679872 : Prime number: 9907
Thread id: 139934389679872 : Prime number: 9923
Thread id: 139934389679872 : Prime number: 9929
Thread id: 139934389679872 : Prime number: 9931
Thread id: 139934389679872 : Prime number: 9941
Thread id: 139934389679872 : Prime number: 9949
Thread id: 139934389679872 : Prime number: 9967
Thread id: 139934389679872 : Prime number: 9973
mahan@mahan-N515-52:~/Documents/Week2$ 
```

1. Convert this program to prompt the user for a number and then to create the number of threads the user has specified to find the prime numbers.

Code:

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

struct threadArgs {
int start;
int finish;
};

void *threadMain(void *p){
struct threadArgs *pargs = p;
int i, c;
int nstart=pargs->start, nfinish=pargs->finish;
pthread_t tid = pthread_self();

for(i=nstart; i<=nfinish; i++){
      for(c=2; c<=i-1; c++) {
      if ( i%c==0 )
      break;
      }
      if ( c==i )
      printf("Thread id: %ld : Prime number: %d\n",tid, i);

      }

}

void main(int argc, char **argv){

int numThreads;

printf("Enter Number of Threads: \n");
scanf("%d",&numThreads);

int i;
pthread_t thrID[numThreads];
struct threadArgs targs[100];

if (numThreads > 0 && numThreads <= 100){
int chunksize = 10000/numThreads ;

for (i=0; i < numThreads; i++){
targs[i].start = i * chunksize;
targs[i].finish = (i * chunksize) + chunksize;
pthread_create(&thrID[i], NULL, threadMain, &targs[i]);
}
for (i=0; i < numThreads; i++){
pthread_join(thrID[i], NULL);
}
}
}
```

Output:



```
Thread id: 140632922453760 : Prime number: 6329
Thread id: 140632914061056 : Prime number: 9241
Thread id: 140632922453760 : Prime number: 6337
Thread id: 140632914061056 : Prime number: 9257
Thread id: 140632922453760 : Prime number: 6343
Thread id: 140632922453760 : Prime number: 6353
Thread id: 140632914061056 : Prime number: 9277
Thread id: 140632922453760 : Prime number: 6359
Thread id: 140632914061056 : Prime number: 9281
Thread id: 140632922453760 : Prime number: 6361
Thread id: 140632922453760 : Prime number: 6367
Thread id: 140632914061056 : Prime number: 9283
Thread id: 140632922453760 : Prime number: 6373
Thread id: 140632914061056 : Prime number: 9293
Thread id: 140632922453760 : Prime number: 6379
Thread id: 140632922453760 : Prime number: 6389
Thread id: 140632914061056 : Prime number: 9311
Thread id: 140632922453760 : Prime number: 6397
Thread id: 140632914061056 : Prime number: 9319
Thread id: 140632922453760 : Prime number: 6421
Thread id: 140632922453760 : Prime number: 6427
Thread id: 140632914061056 : Prime number: 9323
Thread id: 140632922453760 : Prime number: 6449
Thread id: 140632914061056 : Prime number: 9337
Thread id: 140632922453760 : Prime number: 6451
Thread id: 140632922453760 : Prime number: 6469
Thread id: 140632914061056 : Prime number: 9341
Thread id: 140632922453760 : Prime number: 6473
Thread id: 140632914061056 : Prime number: 9343
Thread id: 140632922453760 : Prime number: 6481
Thread id: 140632922453760 : Prime number: 6491
Thread id: 140632914061056 : Prime number: 9349
Thread id: 140632922453760 : Prime number: 6521
Thread id: 140632914061056 : Prime number: 9371
Thread id: 140632922453760 : Prime number: 6529
Thread id: 140632922453760 : Prime number: 6547
Thread id: 140632914061056 : Prime number: 9377
Thread id: 140632922453760 : Prime number: 6551
Thread id: 140632914061056 : Prime number: 9391
Thread id: 140632922453760 : Prime number: 6553
Thread id: 140632922453760 : Prime number: 6563
Thread id: 140632914061056 : Prime number: 9397
Thread id: 140632922453760 : Prime number: 6569
Thread id: 140632914061056 : Prime number: 9403
Thread id: 140632922453760 : Prime number: 6571
Thread id: 140632922453760 : Prime number: 6577
Thread id: 140632914061056 : Prime number: 9413
Thread id: 140632922453760 : Prime number: 6581
Thread id: 140632914061056 : Prime number: 9419
Thread id: 140632922453760 : Prime number: 6599
Thread id: 140632914061056 : Prime number: 9421
Thread id: 140632922453760 : Prime number: 6607
Thread id: 140632922453760 : Prime number: 6619
Thread id: 140632914061056 : Prime number: 9431
Thread id: 140632922453760 : Prime number: 6637
Thread id: 140632914061056 : Prime number: 9433
```

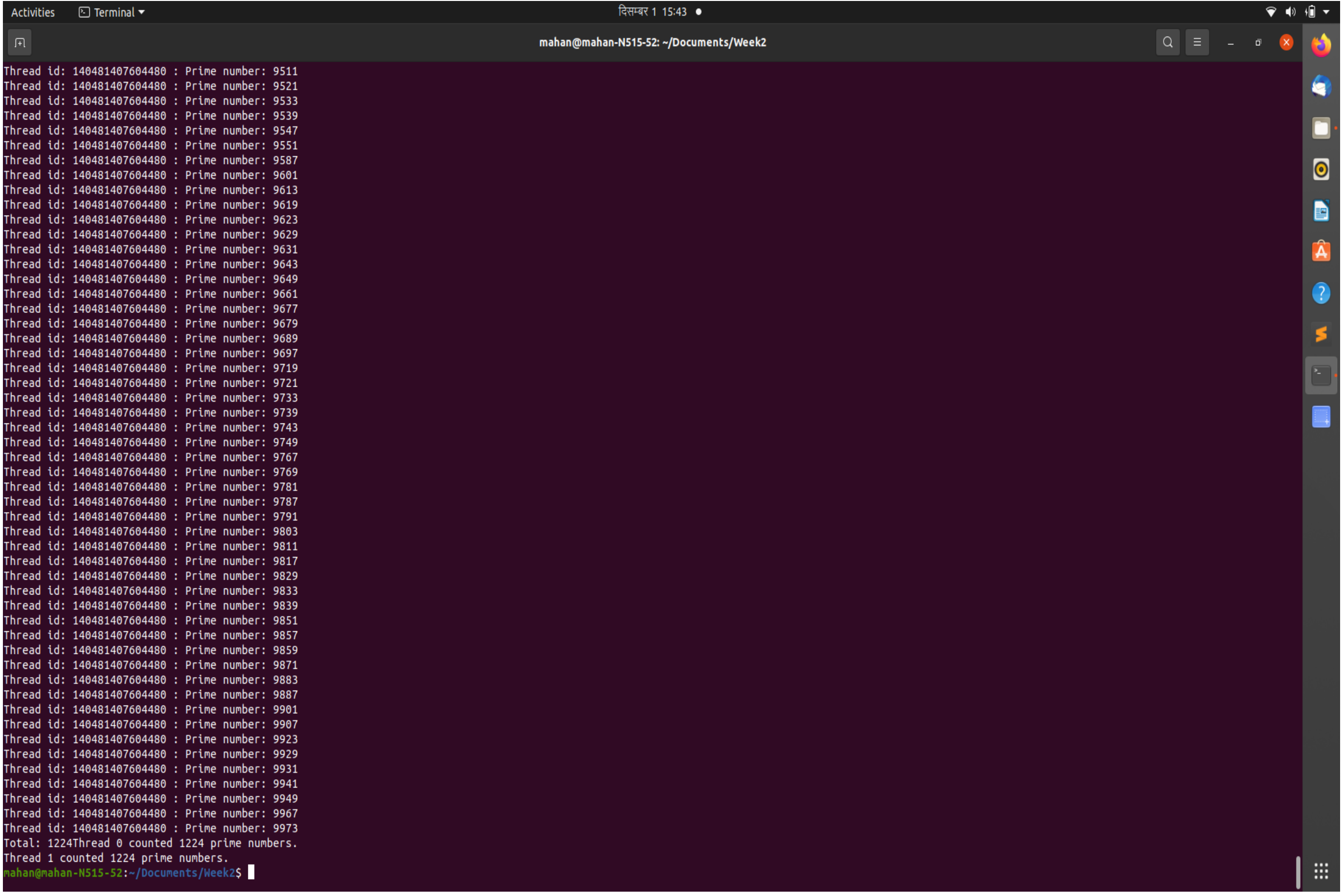
2. Convert the program in (2) so that each thread returns the number of prime numbers that it has found using pthread_exit() and for main program to print out the number of prime number that each thread has found.

Code:

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

struct threadArgs {
int start;
int finish;
};

int count;

void *threadMain(void *p){
struct threadArgs *pargs = p;
int i, c ;
int nstart=pargs->start, nfinish=pargs->finish;
count = 0;
pthread_t tid = pthread_self();

for(i=nstart; i<=nfinish; i++){
      for(c=2; c<=i-1; c++) {
      if ( i%c==0 )
      break;
      }
      if ( c==i ){
      printf("Thread id: %ld : Prime number: %d\n",tid, i);
      count= count+1;
    }
      }
      printf("Total: %d",count);
pthread_exit(&count);
}

void main(int argc, char **argv){


int i;
int numThreads;

printf("Enter Number of Threads: \n");
scanf("%d",&numThreads);

void *totalCount[numThreads];
int total[numThreads];
pthread_t thrID[numThreads];
struct threadArgs targs[100];

if (numThreads > 0 && numThreads <= 100){
```

```
int chunksize = 10000/numThreads ;

for (i=0; i < numThreads; i++){

targs[i].start = i * chunksize;
targs[i].finish = (i * chunksize) + chunksize;
pthread_create(&thrID[i], NULL, threadMain, &targs[i]);
}

for (i=0; i < numThreads; i++){
pthread_join(thrID[i], &totalCount[i]);
}
for (i=0; i < numThreads; i++){
total[i]= *(int *)totalCount[i];
printf("Thread %d counted %d prime numbers. \n",i,total[i]);
}
}
}
```

Output:



```
Thread id: 140481407604480 : Prime number: 9511
Thread id: 140481407604480 : Prime number: 9521
Thread id: 140481407604480 : Prime number: 9533
Thread id: 140481407604480 : Prime number: 9539
Thread id: 140481407604480 : Prime number: 9547
Thread id: 140481407604480 : Prime number: 9551
Thread id: 140481407604480 : Prime number: 9587
Thread id: 140481407604480 : Prime number: 9601
Thread id: 140481407604480 : Prime number: 9613
Thread id: 140481407604480 : Prime number: 9619
Thread id: 140481407604480 : Prime number: 9623
Thread id: 140481407604480 : Prime number: 9629
Thread id: 140481407604480 : Prime number: 9631
Thread id: 140481407604480 : Prime number: 9643
Thread id: 140481407604480 : Prime number: 9649
Thread id: 140481407604480 : Prime number: 9661
Thread id: 140481407604480 : Prime number: 9677
Thread id: 140481407604480 : Prime number: 9679
Thread id: 140481407604480 : Prime number: 9689
Thread id: 140481407604480 : Prime number: 9697
Thread id: 140481407604480 : Prime number: 9719
Thread id: 140481407604480 : Prime number: 9721
Thread id: 140481407604480 : Prime number: 9733
Thread id: 140481407604480 : Prime number: 9739
Thread id: 140481407604480 : Prime number: 9743
Thread id: 140481407604480 : Prime number: 9749
Thread id: 140481407604480 : Prime number: 9767
Thread id: 140481407604480 : Prime number: 9769
Thread id: 140481407604480 : Prime number: 9781
Thread id: 140481407604480 : Prime number: 9787
Thread id: 140481407604480 : Prime number: 9791
Thread id: 140481407604480 : Prime number: 9803
Thread id: 140481407604480 : Prime number: 9811
Thread id: 140481407604480 : Prime number: 9817
Thread id: 140481407604480 : Prime number: 9829
Thread id: 140481407604480 : Prime number: 9833
Thread id: 140481407604480 : Prime number: 9839
Thread id: 140481407604480 : Prime number: 9851
Thread id: 140481407604480 : Prime number: 9857
Thread id: 140481407604480 : Prime number: 9859
Thread id: 140481407604480 : Prime number: 9871
Thread id: 140481407604480 : Prime number: 9883
Thread id: 140481407604480 : Prime number: 9887
Thread id: 140481407604480 : Prime number: 9901
Thread id: 140481407604480 : Prime number: 9907
Thread id: 140481407604480 : Prime number: 9923
Thread id: 140481407604480 : Prime number: 9929
Thread id: 140481407604480 : Prime number: 9931
Thread id: 140481407604480 : Prime number: 9941
Thread id: 140481407604480 : Prime number: 9949
Thread id: 140481407604480 : Prime number: 9967
Thread id: 140481407604480 : Prime number: 9973
Total: 1224Thread 0 counted 1224 prime numbers.
Thread 1 counted 1224 prime numbers.
mahan@mahan-N515-52:~/Documents/Week2$ 
```

4) Convert the program in (3) to use pthread_cancel() to cancel all threads as soon as the 5th prime number has been found.

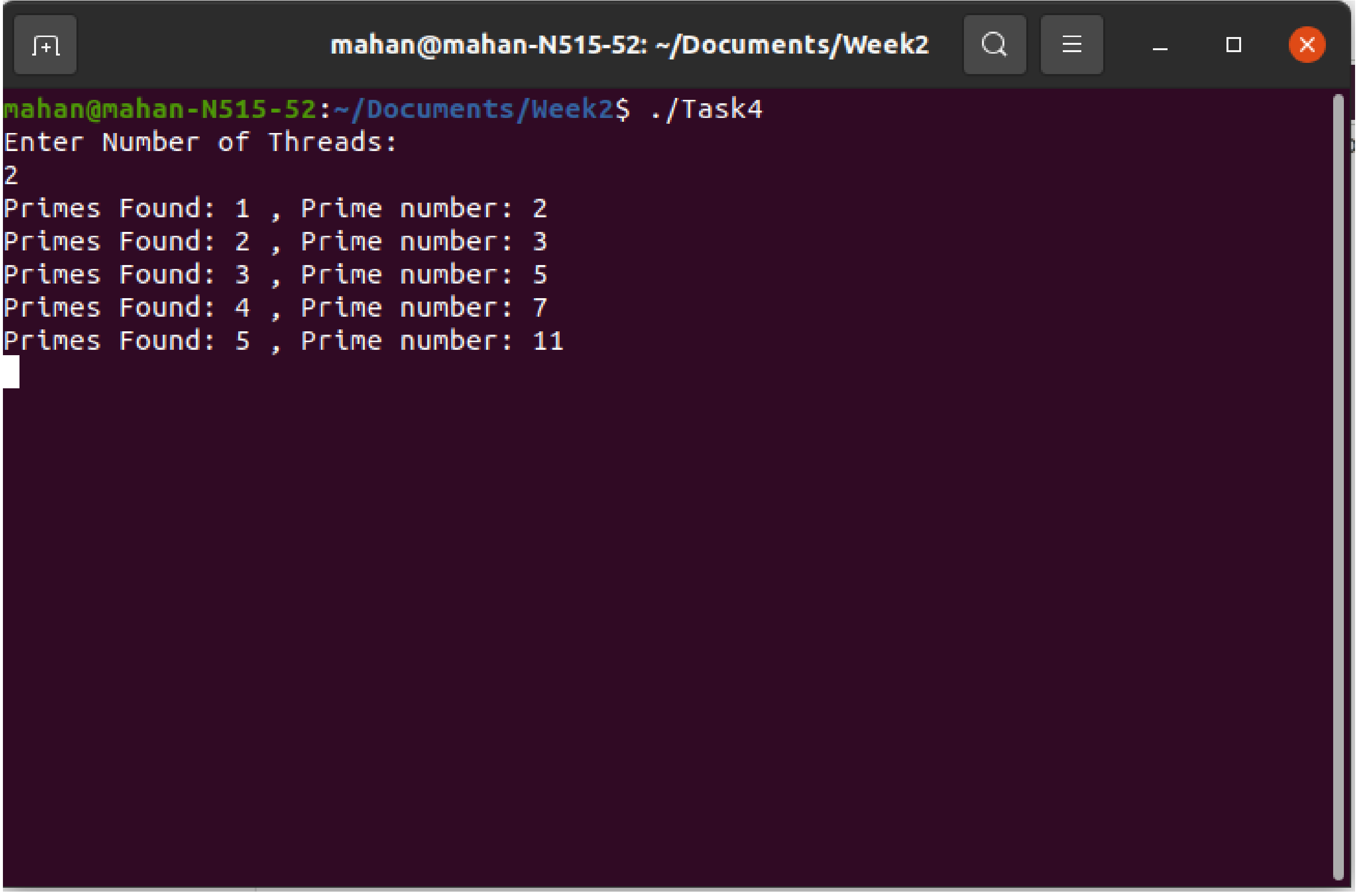
Code:

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


int flag=0;
int primeCount;
pthread_mutex_t lock;


struct threadArgs {
int start;
int finish;
};

void *threadMain(void *p)
{
   pthread_t tid = pthread_self();

 if(primeCount==5){
                        pthread_cancel(tid);
  }

struct threadArgs *pargs = p;
int nstart=pargs->start, nfinish=pargs->finish;
int i,c;

for( i=nstart; i<=nfinish; i++){

                        for( c=2; c<=i-1; c++) {



                                    if ( i%c==0 )
                                          break;
                                                }

         pthread_mutex_lock(&lock);
                                    if ( c==i ){

                                                            primeCount = primeCount + 1 ;
                                        printf("Primes Found: %d , Prime number: %d\n",primeCount,c);

                 }

                        pthread_mutex_unlock(&lock);
```

```
                    if(primeCount==5){
                                        flag =1;
                                        pthread_cancel(tid);
                       }

            }

            pthread_exit(NULL);

          }



void main(){

pthread_mutex_init(&lock,NULL);
int i;
int numThreads;


printf("Enter Number of Threads: \n");
scanf("%d",&numThreads);

void *totalCount[numThreads];

int total[numThreads];


pthread_t thrID[numThreads];

struct threadArgs targs[100];

if (numThreads > 0 && numThreads <= 100){

int workload = 10000/numThreads ;

for (i=0; i < numThreads; i++){

targs[i].start = i * workload;

targs[i].finish = (i * workload) + workload;

pthread_create(&thrID[i], NULL, threadMain, &targs[i]);

}

for (i=0; i < numThreads; i++){
```

```
pthread_join(thrID[i], &totalCount[i]);

}
 printf("%d prime numbers found. \n",primeCount);
}
else{
        printf("Enter a value between 0-101");
}

pthread_mutex_destroy(&lock);


}
```

Output:



```
mahan@mahan-N515-52:~/Documents/Week2$ ./Task4
Enter Number of Threads:
2
Primes Found: 1 , Prime number: 2
Primes Found: 2 , Prime number: 3
Primes Found: 3 , Prime number: 5
Primes Found: 4 , Prime number: 7
Primes Found: 5 , Prime number: 11
```