CSE 4034 – Advanced Unix Programming
Programming Assignment # 2

MEHMET MUM - 150114051
OĞUZHAN BÖLÜKBAŞ - 150114022

Design;

1. Get a number from user.
2. Calculate number of threads. (random between 3 and 5)
3. Allocate memory for the array and negate it.
4. Check primeness of numbers which are less then the number.
   - Divide numbers to previous prime numbers. If the number is not divisible none of them, the number is prime.
5. Print prime numbers. (in the pdf file it does not say threads have to print their prime numbers. We recorded prime numbers and which thread calculated it to a array, and after all calculations are finished, main thread prints the prime numbers and threads.)

Number : 50
Number of working threads : 4

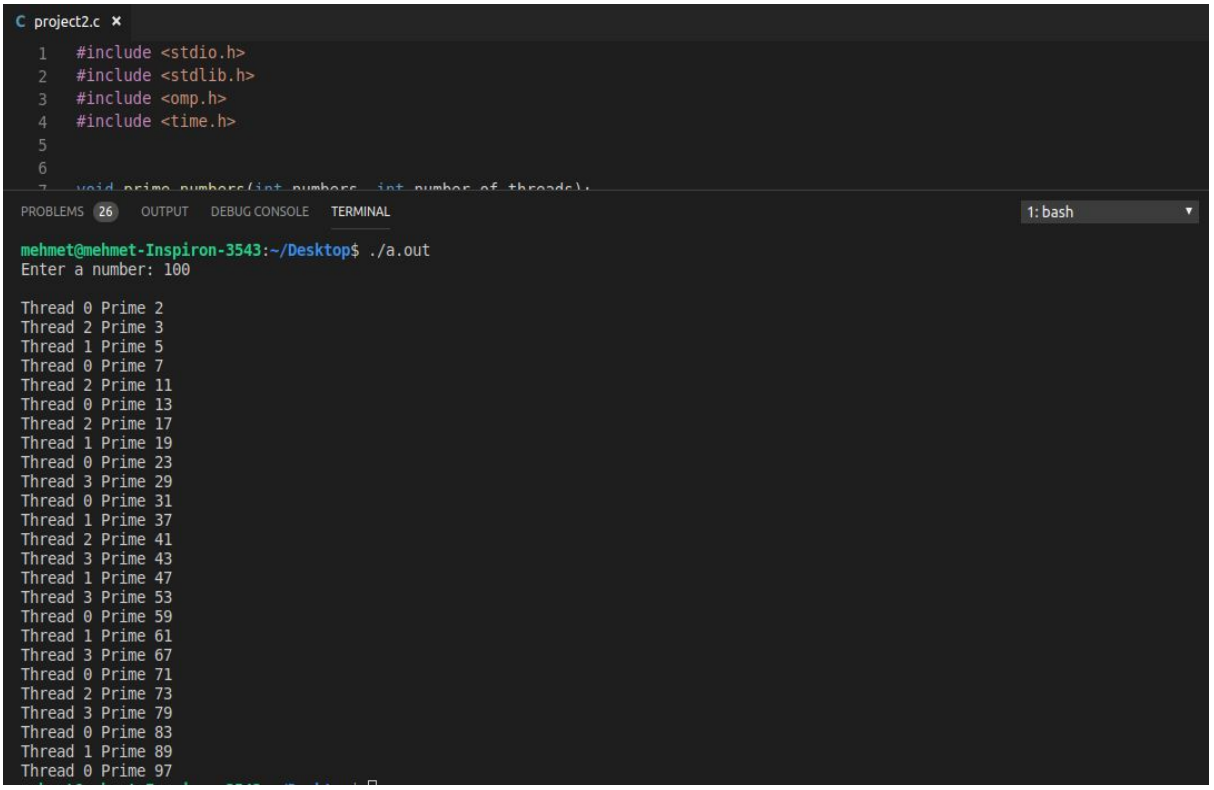
project2.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
#include <time.h>


void prime_numbers(int numbers, int number_of_threads);
int check_prime(int the_number);
void print_primes(int number);
void negate(int number);
void memory_allocate(int number);

int **prime_array;      // array for keep prime numbers  and  which thread calculated it
// prime_array[i][0]  keeps prime or not
// prime_array[i][1]  keeps which thread calculate it
```

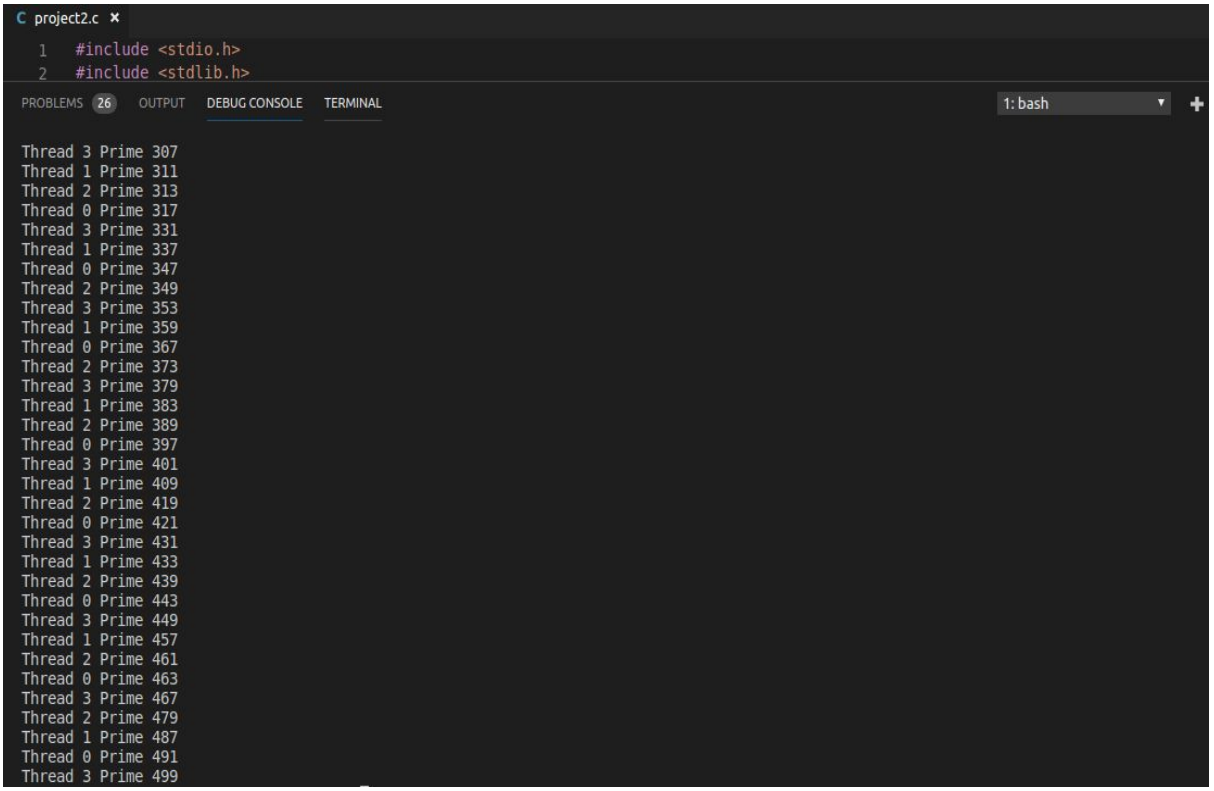
PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL 1: bash

```
mehmet@mehmet-Inspiron-3543:~/Desktop$ ./a.out
Enter a number: 50

Thread 0 Prime 2
Thread 2 Prime 3
Thread 1 Prime 5
Thread 0 Prime 7
Thread 1 Prime 11
Thread 2 Prime 13
Thread 0 Prime 17
Thread 3 Prime 19
Thread 0 Prime 23
Thread 3 Prime 29
Thread 0 Prime 31
Thread 3 Prime 37
Thread 1 Prime 41
Thread 2 Prime 43
Thread 3 Prime 47
```

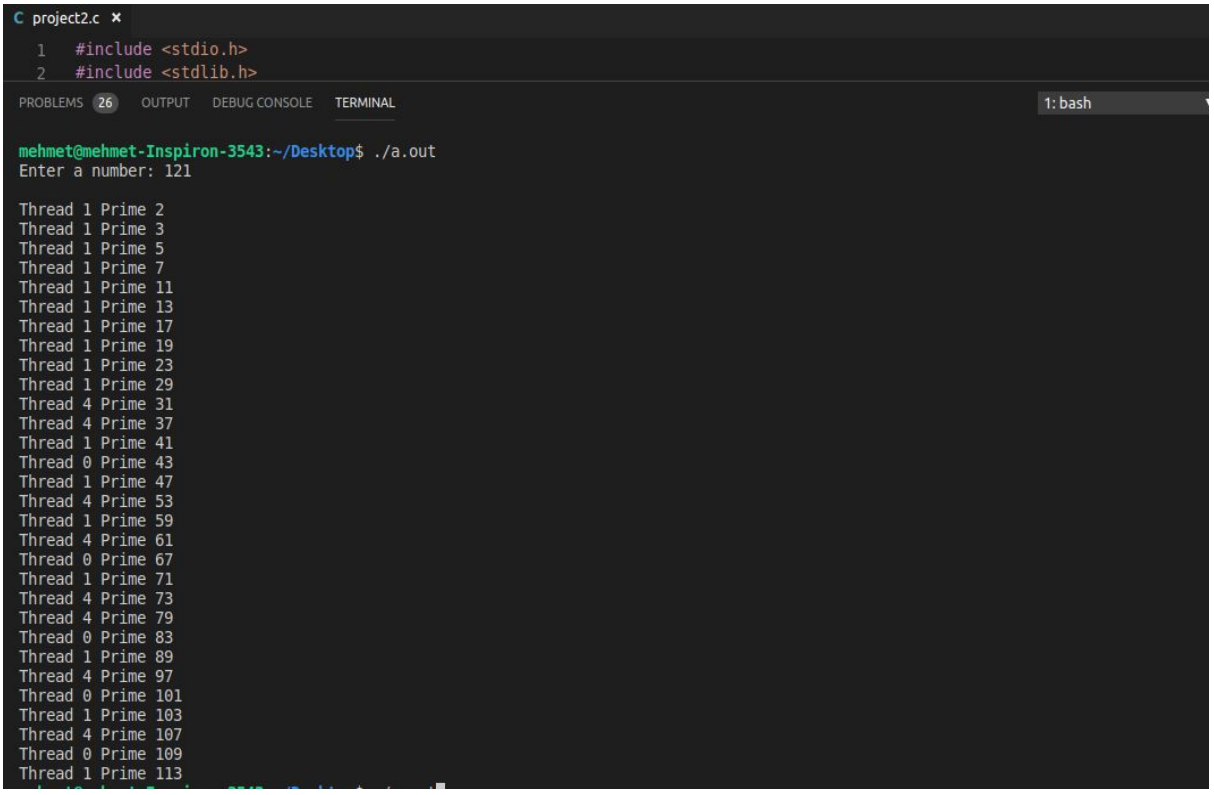
Number : 100
Number of working threads : 4



Number : 500
Number of working threads : 4



Number : 121
Number of working threads : 5



Number : 70
Number of working threads : 3