

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

```
C project2.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <omp.h>
4  #include <time.h>
5
6
7  void prime_numbers(int numbers, int number_of_threads);
8  int check_prime(int the number);
9  void print_primes(int number);
10 void negate(int number);
11 void memory_allocate(int number);
12
13 int **prime_array;    // array for keep prime numbers and which thread calculated it
14 // prime_array[i][0] keeps prime or not
15 // 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

```
C project2.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <omp.h>
4  #include <time.h>
5
6
7  void prime_numbers(int numbers, int number_of_threads);

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL 1: bash
mehmet@mehmet-Inspiron-3543:~/Desktop$ ./a.out
Enter a number: 100

Thread 0 Prime 2
Thread 2 Prime 3
Thread 1 Prime 5
Thread 0 Prime 7
Thread 2 Prime 11
Thread 0 Prime 13
Thread 2 Prime 17
Thread 1 Prime 19
Thread 0 Prime 23
Thread 3 Prime 29
Thread 0 Prime 31
Thread 1 Prime 37
Thread 2 Prime 41
Thread 3 Prime 43
Thread 1 Prime 47
Thread 3 Prime 53
Thread 0 Prime 59
Thread 1 Prime 61
Thread 3 Prime 67
Thread 0 Prime 71
Thread 2 Prime 73
Thread 3 Prime 79
Thread 0 Prime 83
Thread 1 Prime 89
Thread 0 Prime 97
```

Number : 500

Number of working threads : 4

```
C project2.c x
1  #include <stdio.h>
2  #include <stdlib.h>
```

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL

1: bash

```
Thread 3 Prime 307
Thread 1 Prime 311
Thread 2 Prime 313
Thread 0 Prime 317
Thread 3 Prime 331
Thread 1 Prime 337
Thread 0 Prime 347
Thread 2 Prime 349
Thread 3 Prime 353
Thread 1 Prime 359
Thread 0 Prime 367
Thread 2 Prime 373
Thread 3 Prime 379
Thread 1 Prime 383
Thread 2 Prime 389
Thread 0 Prime 397
Thread 3 Prime 401
Thread 1 Prime 409
Thread 2 Prime 419
Thread 0 Prime 421
Thread 3 Prime 431
Thread 1 Prime 433
Thread 2 Prime 439
Thread 0 Prime 443
Thread 3 Prime 449
Thread 1 Prime 457
Thread 2 Prime 461
Thread 0 Prime 463
Thread 3 Prime 467
Thread 2 Prime 479
Thread 1 Prime 487
Thread 0 Prime 491
Thread 3 Prime 499
```

Number : 121

Number of working threads : 5

```
C project2.c x
1 #include <stdio.h>
2 #include <stdlib.h>

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL 1: bash

mehmet@mehmet-Inspiron-3543:~/Desktop$ ./a.out
Enter a number: 121

Thread 1 Prime 2
Thread 1 Prime 3
Thread 1 Prime 5
Thread 1 Prime 7
Thread 1 Prime 11
Thread 1 Prime 13
Thread 1 Prime 17
Thread 1 Prime 19
Thread 1 Prime 23
Thread 1 Prime 29
Thread 4 Prime 31
Thread 4 Prime 37
Thread 1 Prime 41
Thread 0 Prime 43
Thread 1 Prime 47
Thread 4 Prime 53
Thread 1 Prime 59
Thread 4 Prime 61
Thread 0 Prime 67
Thread 1 Prime 71
Thread 4 Prime 73
Thread 4 Prime 79
Thread 0 Prime 83
Thread 1 Prime 89
Thread 4 Prime 97
Thread 0 Prime 101
Thread 1 Prime 103
Thread 4 Prime 107
Thread 0 Prime 109
Thread 1 Prime 113
```

Number : 70

Number of working threads : 3

```
C project2.c •
83 }
84
85 // print prime numbers
86 void print_primes(int number){
87     int i;
88     for (i = 0; i<number; i++){
89         if ( prime_array[i][0] == 1)
90             printf("Thread %d Prime %d\n", prime_array[i][1], i);
91     }
92 }

PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL 1: bash

Thread 1 Prime 47
Thread 0 Prime 53
Thread 3 Prime 59
Thread 1 Prime 61
Thread 2 Prime 67
mehmet@mehmet-Inspiron-3543:~/Desktop$ ./a.out
Enter a number: 70

Thread 1 Prime 2
Thread 0 Prime 3
Thread 1 Prime 5
Thread 2 Prime 7
Thread 2 Prime 11
Thread 0 Prime 13
Thread 1 Prime 17
Thread 2 Prime 19
Thread 2 Prime 23
Thread 1 Prime 29
Thread 2 Prime 31
Thread 1 Prime 37
Thread 2 Prime 41
Thread 0 Prime 43
Thread 2 Prime 47
Thread 1 Prime 53
Thread 0 Prime 59
Thread 1 Prime 61
Thread 2 Prime 67
```