

Summer Term COM1002 Computer Programming II

Lab 2 Q2

Name your files as studentNo.c (12345678.c). Do not make any additions or deficiencies in the outputs (outputs produced by the programs) other than those requested from you. Do not make any changes in outputs such as extra spaces, going to the bottom line. Your codes will be passed through the copy program and the legal process will be initiated against the student who have similarity rates higher than the determined rate. For this reason, the assignments you upload should be unique to you.

You can compile your codes as follows:

```
gcc studentNumber.c  
./a.out
```

Make sure your program is running in Ubuntu environment. Do not forget that there may be programs that run on a different operating system but generate errors or warnings in Ubuntu. Objections to the evaluations made due to such errors will not be accepted. Please carefully review the sample input and output files provided for the correct output format.

To check the accuracy of your program, you need to perform the following operations.

1) gcc studentNumber.c

```
./a.out> myOutput1.txt
```

This command saves the output of your program to myOutput1.txt file.

2) diff myOutput1.txt output1.txt

Using this command, compare your own output with the required output. If there is no warning on the screen after entering this command, your program is working correctly for these values. If you see a warning in the command system after entering the command, this indicates a problem with your output, you need to correct your code.

Test your program for different inputs that you will create yourself. The input files given to you and the input files used during the evaluation will differ.

Q2.(60 minutes)

As you know prime numbers are not spaced in a regular way which means there is a different distance between them. For example the distance between

2 and 3 is 1

3 and 5 is 2

7 and 11 is 4

You will write a C program which prints the first pair of two prime numbers which has distance d with the parameters;

distance, d,

first integer, (a) starting point

second integer, (b) end point

function prototype is

```
void distance_between_prime(d, a, b)
```

Sample input output will be like:

`distance_between_prime (2, 3, 50)` —> 3 and 5

There are many pairs with distance 2 between the number 3 and 50

like 3-5, 5-7, 11-13, 17-19

but the first one is 3 and 5

`distance_between_prime (4, 130, 200)` —> 163 and 167