CSE102 – Computer Programming (Spring 2022)

Homework #3

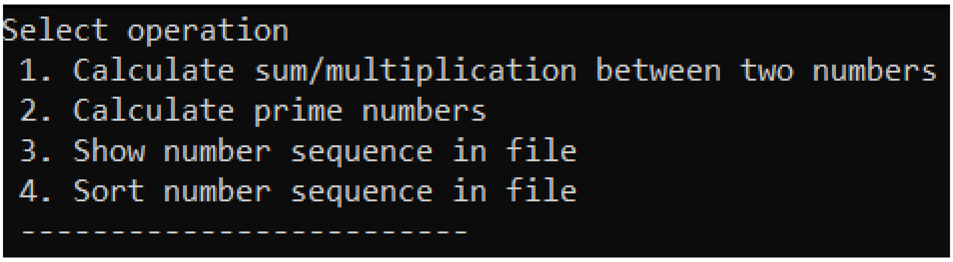
Handed out: March 21, 2022.

Due: 11:55pm March 30, 2022.

Hand-in Policy: Via Teams. No late submissions will be accepted. Files named main.c and makefile for this homework and compress it into a StudentNumber_Name_Surname.zip file. You can email seydanurahi@gtu.edu.tr if you have any questions.
Collaboration Policy: No collaboration is permitted.
Grading: This homework will be graded on the scale of 100.

Homework Description: Write a complete program that performs following tasks respectively. These tasks should be distinct functions and called from main() function with selection. Once they been called, each will call another function to make the calculation for the related task. These calls will be made by selecting the inputs received from the user. For each process, the input / output part and the calculation part will be created separately. So you have to do the calculations in a separate function. This function should take the values required for calculation from the previous function as input and return output as a result. You should create a menu as shown below.

Expected Menu



```
Select operation
 1. Calculate sum/multiplication between two numbers
 2. Calculate prime numbers
 3. Show number sequence in file
 4. Sort number sequence in file
 -------------------------
```

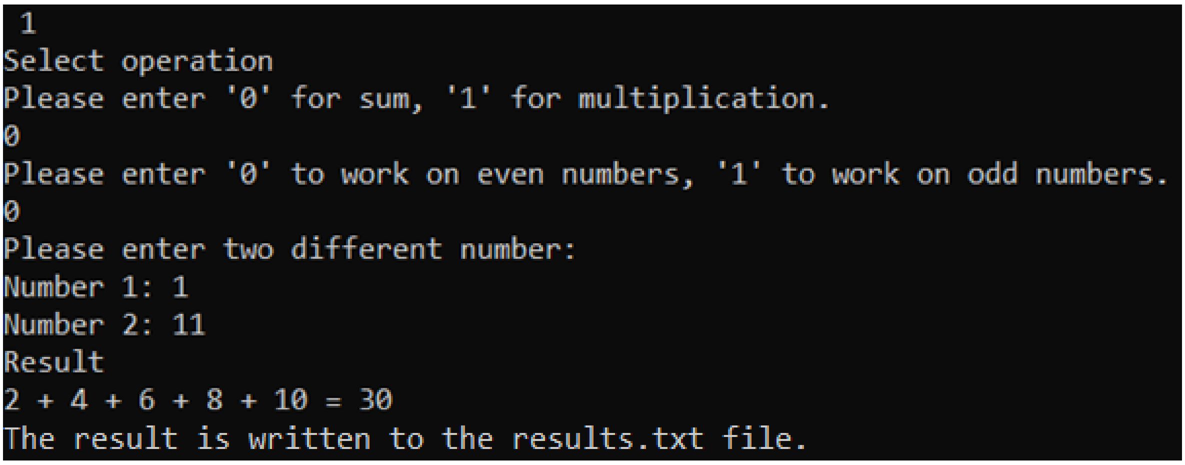
Part 1. [25 pts] In this function, the program respectively request a two Integer (N1, N2), an operation flag, and finally a flag to decide if it is going to work on odd / even numbers. The program will support only 2 operations, addition or multiplication. According to the selection of the flags, the program will calculate the sum/multiplication of the odd/even numbers between the range of [N1,N2]. The addition and the multiplication operations should be done by using two different functions. The operation selection should be determined by using operation flag with switch-case, after that, the integer and the other flag should be used to call the related function. If user enters invalid value for the flags, the program should print an error message. After each calculation process is completed, the result should be written to the file named "results.txt".

Function prototypes are :

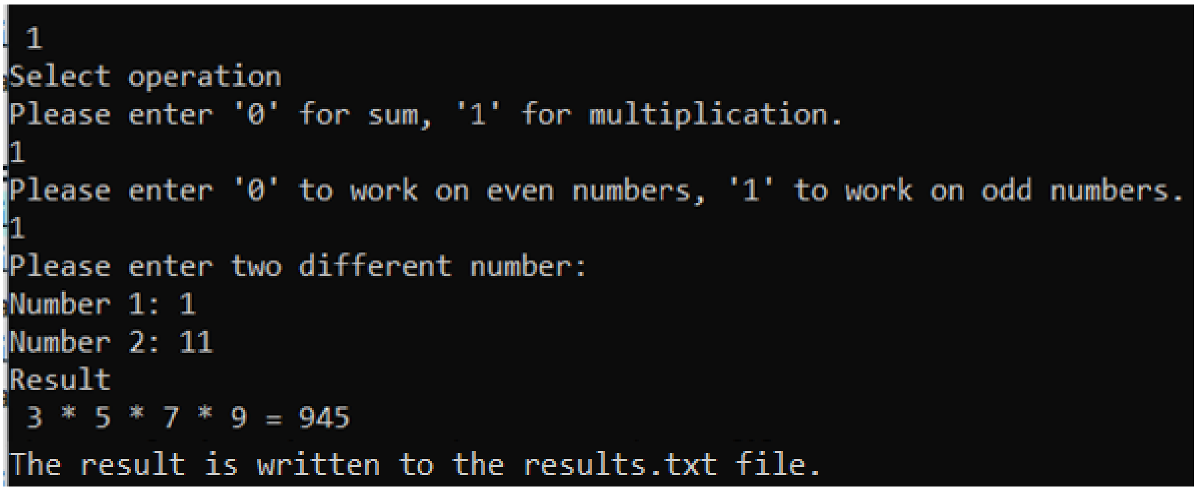
int sum (int n1, int n2, int flag)

int multi (int n1, int n2, int flag)

Examples:



```
 1
Select operation
Please enter '0' for sum, '1' for multiplication.
0
Please enter '0' to work on even numbers, '1' to work on odd numbers.
0
Please enter two different number:
Number 1: 1
Number 2: 11
Result
2 + 4 + 6 + 8 + 10 = 30
The result is written to the results.txt file.
```



```
 1
Select operation
Please enter '0' for sum, '1' for multiplication.
1
Please enter '0' to work on even numbers, '1' to work on odd numbers.
1
Please enter two different number:
Number 1: 1
Number 2: 11
Result
 3 * 5 * 7 * 9 = 945
The result is written to the results.txt file.
```

Part 2. [25 pts] Write a function which takes an integer N from the user and checks the every integer from 2 to that number if they are prime or not. The primality testing is made by a function with the following information:

A is prime if A is not dividible by any integer X where X is;

$$1 < X \leq \sqrt{A}$$

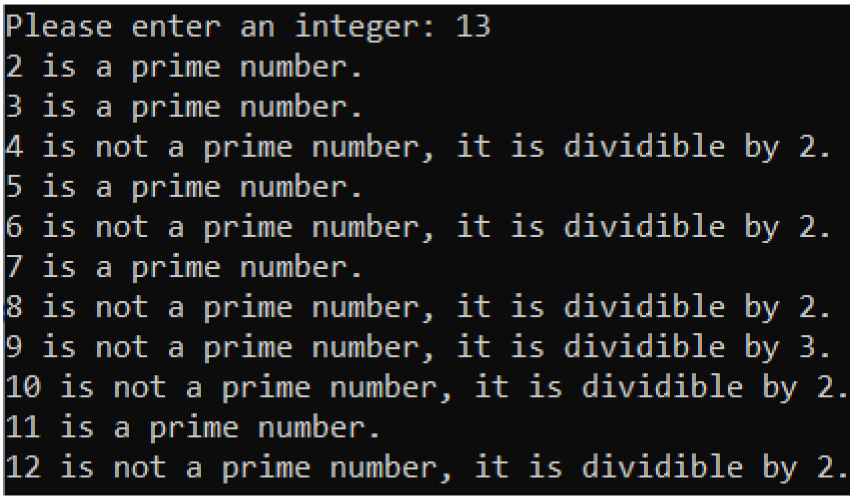
This operation should be done in a function, the function should return a flag if the integer is prime, or should return the least divisor of that integer if it is not a prime. In the main function, you should use a loop to check every A between $1 < A < N$ obtain a result by using the function and finally print it.

Only 'for' loops should be used. You are allowed to use sqrt() function from math library to calculate the square root.

Function prototype is :

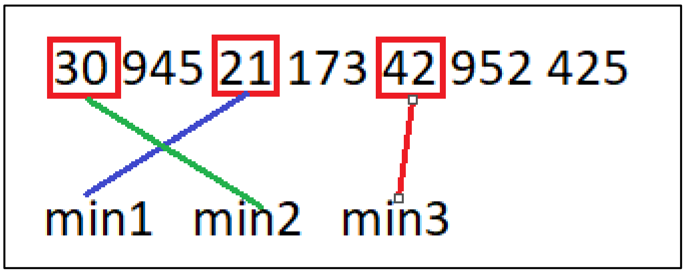
int isprime (int a)

Expected Output



```
Please enter an integer: 13
2 is a prime number.
3 is a prime number.
4 is not a prime number, it is dividible by 2.
5 is a prime number.
6 is not a prime number, it is dividible by 2.
7 is a prime number.
8 is not a prime number, it is dividible by 2.
9 is not a prime number, it is dividible by 3.
10 is not a prime number, it is dividible by 2.
11 is a prime number.
12 is not a prime number, it is dividible by 2.
```

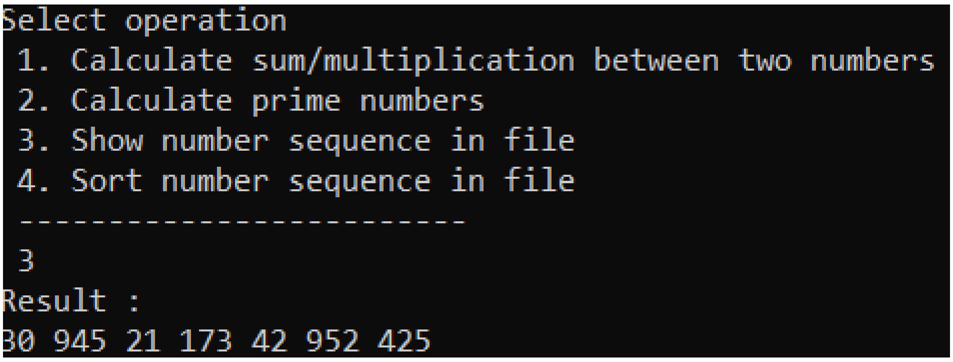
Part 3. [50 pts]: In file operations, you must create a function named write_file() for writing to the file. This function should take the value to be written to the file as a parameter. The 3rd option on the menu should call the print_file() function and print the numbers in the "results.txt" to the terminal. The 4th option in the menu should call the function named sort_file(). In this function, a simple sorting operation should be performed in ascending order in the "results.txt" file. In this sorting process, you need to sort the mixed numbers in the results.txt file from smallest to largest. You should select 3 numbers in each step. You should determine the smallest three numbers in sequence and keep them on a temporary file. In other words, you should assign the smallest numbers to the min1, min2, min3 variables and keep them in order from smallest to largest in a temporary file. A sample image of this is below. In each sorting round, the smallest 3 numbers should be determined and transferred to the temporary file. You are not allowed to use arrays. You can create the necessary functions by adding your comment lines. When you complete all the operations, you should print all the numbers in order from smallest to largest on "results.txt". Remember, you need to sort the numbers 3 by 3, not one by one.



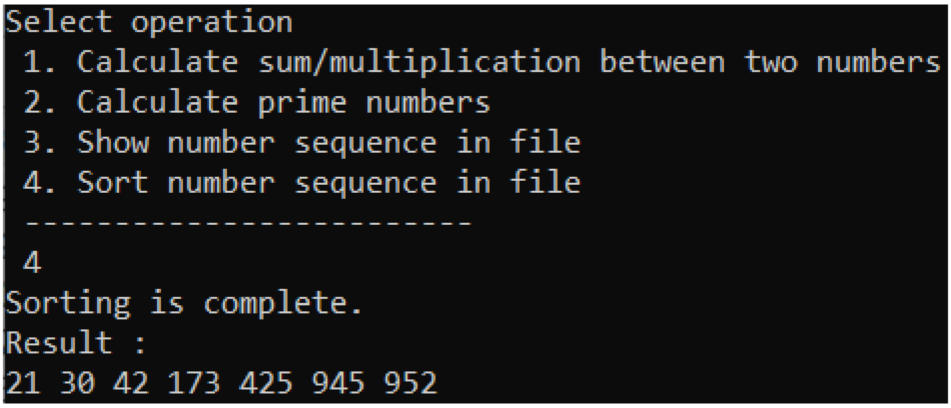
Function prototypes are:

```
void write_file (int number)

void sort_file ()

void print_file ()
```

Expected Outputs :



```
Select operation
 1. Calculate sum/multiplication between two numbers
 2. Calculate prime numbers
 3. Show number sequence in file
 4. Sort number sequence in file
 -------------------------
 3
Result :
30 945 21 173 42 952 425
```



```
Select operation
 1. Calculate sum/multiplication between two numbers
 2. Calculate prime numbers
 3. Show number sequence in file
 4. Sort number sequence in file
 -------------------------
 4
Sorting is complete.
Result :
21 30 42 173 425 945 952
```

General Rules:

1. Obey the style guidelines.
2. Do not change the provided function prototypes (you will not get any credits).
3. The program must be developed on Ubuntu using GCC compiler (version provided in class), compilation problems due to the use of another OS or compiler is your responsibility (you will not get any credits).
4. Your program should work as expected. Do not expect partial credit if your code works only in some cases but not in all cases as expected.
5. You can ask your questions about the homework by posting on the forum in Teams. Handing in your work:
6. Hand in your work using the appropriate class Teams assignment site.
7. No late submissions will be accepted.
8. Pack this directory into a zip file named 20180000001_X_Z.zip
9. When unpacked as above in Ubuntu (version provided in class) it should allow executing the following commands in a shell:
   ▪ “$make clean” removes everything except makefile, source code (.c) and other resource files (if any) – all compiling results and intermediate files should be removed (except results.txt).
   ▪ “$make compile” should compile the code.
   ▪ “$make run” should run the code along with any parameters needed.