

CSCI-UA.0201
Computer Systems Organization
Programming Assignment #1

In this first programming assignment you will write C code. You have six C assignment to solve all included in the same C file: lab1.c

This means all your work will be inside this C file: lab1.c.

The six assignments are described below. But before you start coding, please read the following list of bullets first.

- Read ALL the comments on the C file before you write anything.
- It will help you a lot if you read and understand the main() function.
- Include your name and NetID in the corresponding comment at the top of the file.
- Do all the work in the virtual machine. Do NOT try to do it on your MAC or Windows and then move it to the machine. Many things can break.
- Do not change any code in the file where it states so. So do not change function declarations or the main function.
- However, you are free to add extra functions if you want.

You compile your code as:

gcc -Wall -o lab1 lab1.c

Here are the descriptions of the assignments. You will also find more info in the comments on top of each function the C file.

Insertion sort [10 points]:

In this function, the user enters the number of elements X. The main() function will generate a list of X random numbers. This list together with X are passed to your function insertion_sort() where you have to sort the list in ascending order using the insertion sort algorithm. If you do not remember the insertion sort algorithm, here is a quick description.

The algorithm takes one number at a time and puts it in its correct place in the list. For example, if we have a list: { 4 3 1 }, the algorithm will scan the list, pick one number at a time, and insert it in its right place of the sorted list. We have 3 numbers above, so they will be sorted in 3 steps:

Step 1: 4

Step 2: 3 4

Step 3: 1 3 4

Prime numbers [5 points]:

Your second function is called `find_prime()`. It takes two positive numbers x and y where $x < y$. The function must print all the prime numbers between x and y (not including x and y themselves).

Flipping [5 points]:

This function reads a integer and prints it in reverse order. So, if the user enters 12345, your function must print: 54321.

Histogram [5 points]:

This function reads a text file that contains only lower case alphabet characters. Then prints on the screen a histogram of the characters. For example, if the file contains: aaabbc the function must print on the screen:

a: 3

b: 2

c: 1

d: 0

e:0

.

.

.

From Lower Case to Upper [5 points]:

In this part of the assignment, you design a function that reads a text file containing only lower case characters. It generates a different file, with a different name as indicated in the comments in `lab1.c`, that contains the same characters but in upper case.

Encryption [10 points]:

Here, you will read a text file and print on the screen an encrypted version of that file, as follows:

- Each character will be replaced with another character 3 letters before. For example: f is replaced with c, h is replaced with e, and so on.
- The encryption is circular, so a is replaced with x, b with y, and so on.
- White spaces are ignored and printed unchanged.

Final note:

The main philosophy of this programming assignment is to get you to try several things. You have to read a code and understand it (the code of the main function and the function declarations). You have to implement several functions that touch upon many concepts of C.

So

HAVE FUN!