

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 -std=c99 -o lab1 lab1.c
```

You then execute the lab by typing:

```
./lab1 x [input]
```

Where:

x can take the value of 1 to 6, corresponding to each one of the assignments below. [input] the argument of the corresponding assignment if any. Some assignments do not need arguments.

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

1. Number Arrangement [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 arrange() where you have to arrange the elements as follows.

First, the even elements, if any, are inserted first and sorted in ascending order. Then, the odd numbers, if any, are inserted and sorted in ascending order too. Note that numbers can be repeated.

For example, if we have a list of 5 elements: 5 5 4 3 2 3 1

The output of your function, printed on the screen, will be: 2 4 1 3 3 5 5

2. Fibonacci sequence [10 points]:

Your second function is called `find_fibonacci()`. It takes two positive numbers x and y where $x < y$. The function must print all the Fibonacci series between x and y (not including x and y). Fibonacci is a series of numbers in which each number is the sum of the two preceding numbers. It starts as: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, etc.

So, if $x = 5$ and $y = 20$ then your function must print on the screen: 8 13.

Both x and y will be positive numbers less than 1,000,000 (one million).

3. Flipping [10 points]:

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

4. Histogram [10 points]:

This function reads a text file that contains a string of characters (can be upper case letters, lower case letters, numbers, space, ...). Then your function prints on the screen a histogram of the lower case characters ONLY. For example, if the file contains:

```
AAaaa77b bc
```

the function must print on the screen:

```
a: 3
```

```
b: 2
```

```
c: 1
```

```
d: 0
```

```
e: 0
```

```
.
```

```
z: 0
```

NOTE: Your function has to print ALL the lower case alphabet, even if a character never appears in the file.

5. 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.

6. Encryption [5 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.
- The input file will contain nothing but lower case characters and spaces (no digits, no upper case, etc).
- The output is printed on the screen.

Final notes:

- 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.
- Please, write as many comments as possible with your code so that we can give you partial credit in case your code has something wrong.
- You must not alter the declarations of the functions originally in lab1.c but feel free to add any extra functions you want.

And

HAVE FUN!