

Homework Assignment #0 - Getting Familiar with C++ and Coding on Linux

- Type: Individual, not group allowed
- Synopsis: Implementing standard operations in C++
- Keywords: File I/O, Dynamic Memory Allocation
- Requirements: You may code on your own PC but you have to make sure your program compiles and runs correctly on general.asu.
- Deliverables: A zip/tar archive containing the following:
 - **main.cpp**
 - **Makefile**
 - A **README.txt** with your name in it that documents what your program is doing
 - ** You DO NOT need to turn in sub.o, input.txt and output.txt since they will be replaced anyways during grading
- Due: Wed, Jul 6th at 2359 hours.

Instructions

1. Download the starter package from Canvas, you are given 2 template files: main.cpp, Makefile, and 3 test objects that you should not temper with: sub.h, sub.o, input.txt .

sub.h and sub.o contain the header and the binary of a pre-built function ready to use:

```
int sub(int n, int *A, int *B, int *C);
```

While a sample `input.txt` always looks like this:

```
5
1 2 3
3 4 5
5 6 7
7 8 9
9 10 11
```

The first line defines the number of lines that follow. Each following line always contains 3 integers, separated by whitespaces.

2. In `main.cpp`, implement the following steps:
 - a. Load the first line of `input.txt` into an integer variable `n`;
 - b. Dynamically initialize three integer arrays of size `n`: `A`, `B`, and `C`;

- c. Load each column starting from the second line of `input.txt` respectively into A, B, and C. E.g. A will become [1, 3, 5, 7, 9], B = [2, 4, 6, 8, 10], and likewise C = [3, 5, 7, 9, 11].
 - d. Call the sub function with
`sub(n, A, B, C);`
and write the function's result into a new file `output.txt`.
3. Change the given `Makefile` so that your code compiles into a working binary program that executes the implementations above.

Rubrics

We will measure your performance based on the following checklist:

1. You have turned in an appropriate Documentation readme. (1pt)
2. You have turned in an appropriately modified Makefile. (1pt)
3. Your code generates correct outputs with the grader's own test input files. (2pt)
4. Your code handles File I/O and Dynamic Memory Allocation correctly. (1pt)

We will build your program with the Makefile you turn in. E.g. If your code fails to build on the general.ASU server, then you will get at most (1/5).

What to Turn In, One More Time

- A zip/tar archive containing the following:
 - **main.cpp**
 - **Makefile**
 - A **README.txt** with your name in it that documents what your program is doing
 - ** You DO NOT need to turn in sub.o, input.txt and output.txt since they will be replaced anyways during grading

Tutorials You Might Need

- <https://fpsluozi.github.io/Linux-Setup/>
- <https://www.guru99.com/cpp-file-read-write-open.html>
- <https://stackoverflow.com/questions/39523835/how-to-take-multiple-inputs-in-the-same-line-in-c>
- <https://stackoverflow.com/questions/35532427/how-to-dynamically-allocate-arrays-in-c>