CSE310 Project 01: Basics of Linux, Files, Dynamic Memory Allocation, Modular Design, Separate Compilation, and Makefile

OUT: Tuesday, 01/24/2017

DUE: Tuesday, 02/07/2017 (by 4:30pm)

The objectives of this project are making sure you (1) can implement an algorithm on general.asu.edu; (2) can program in C++; (3) know how to use dynamic memory allocation; (4) know how to read from and write to a file; (5) know how to write a program in multiple modules; and (6) can use a simple Makefile.

You should program on general.asu.edu, and use the programming language C++. No other programming language is accepted. **This is an individual project**.

You should assume that there will be an object file named "foreignsub.o" which implements a function whose prototype is given in the header file "foreignsub.h". The header file "foreignsub.h" consists of the following line.

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

However, you do not know what this function does exactly. For your testing purpose, you may produce such a function, e.g., returning the maximum value among the $3n$ integers in the three arrays.

You are also given an ASCII file named "input.txt". The first line of this file contains an integer n. There are n additional lines in the file, where the ith additional line contains three integers x_i, y_i, and z_i. A sample of "input.txt" is the following.

```
5
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15
```

You need to write a driver that does the following

- Open an input file (for reading) named "input.txt". You program should output an error message and stop, if the file cannot be opened for reading.
- Read the first integer in the file into an integer var named n.

- Dynamically allocate memory for an array A of size n, an array B of size n, and an array C of size n.
- Read data from the input file into the arrays, so that $A[i-1] = x_i$, $B[i-1] = y_i$ and $C[i-1] = z_i$ for $i = 1, 2, \ldots, n$.
- Close the file "input.txt".
- Open an output file (for writing) named "output1.txt". You program should output an error message and stop, if the file cannot be opened for writing.
- Write n into the file "output1.txt" (in the first line), followed by a newline. Then write $A[i]$, $B[i]$ and $C[i]$ in the next line, for $i = 0, 1, \ldots, n-1$.
- Close the file "output1.txt".
- Call the function sub by the command

```
result = sub(n, A, B, C);
```

- Open an output file (for writing) named "output2.txt". You program should output an error message and stop, if the file cannot be opened for writing.
- Write n into the file "output2.txt" (in the first line), followed by a newline. Then write $A[i]$, $B[i]$ and $C[i]$ in the next line, for $i = 0, 1, \ldots, n-1$.
- Write the value of result into the file "output2.txt"..
- Close the file "output2.txt".
- Stop.

Besides writing the source code of the driver, you also need to create a makefile named "Makefile". The makefile should produce an executable file named "proj1". You should submit (in one zip file) the file "Makefile", "foreignsub.h", and the source code of your driver, which is "main.cpp". When we grade your work, we will provide the file "foreignsub.o", and type in "make" to compile on general.asu.edu.

Grading policies:

You should use C++ as the programming language. Your program should be working on general.asu.edu. You will need to submit it electronically at the blackboard, in one zip file, named

CSE310-P01-Lname-Fname, where Lname is your last name and Fname is your first name. If you program does not compile on general.asu.edu, you will receive 0 on this project.

(20 pts) Documentation: You should provide sufficient comment about the variables and operations.

(10 pts) Makefile.

(10 pts) File operations.

(10 pts) I/O operations.

(20 pts) Dynamic memory allocation.

(20 pts) Correct calling of the sub function.

(10 pts) Detecting errors.

Note that you are supposed to know how to program in C++. If you forget what you have learned in the prerequisite, you may google "C++ tutorial" and learn by yourself. The basics of makefile will be covered in class. You can also google "Makefile" to learn more.

The following are some helpful hints (google them).

```
FILE *fp;
int *A;

fp = fopen("input.txt", "r");
if (fp == NULL) exit;

fscanf(fp, "%d", &n);
A = (int *) malloc (n * sizeof (int));
fclose(fp);
```

```
run     :main.o sub1.o sub2.o
         g++ -o run main.o sub1.o sub2.o

main.o  :main.cpp sub1.h sub2.h
         g++ -c main.cpp

sub1.o  :sub1.cpp sub1.h
         g++ -c sub1.cpp
```

```
sub2.o  :sub2.cpp sub2.h
         g++ -c sub2.cpp

clean   :
        rm *.o

cleanAll   :
        rm *.o run
```