CIS 330: Project #2B
Assigned: April 18th, 2018
Due April 23rd, 2018
(which means submitted by 6am on April 24th, 2018)
Worth 4% of your grade

Assignment: Write a program that reads the file "2B_binary_file". This file contains a two-dimensional array of 100 integers (indices 0 to 99); that is, 10x10. You are to read in the 5x5 top left corner of the array. That is, the values at indices 0-4, 10-14, 20-24, 30-34, and 40-44. You may only read 25 integers total. Do not read all 100 and throw some out. You will then write out the new 5x5 array to the output file, which is one of the command line arguments. Please write one integer per line (25 lines total). You should be able to "cat" the file afterwards and see the values.

Use only C file stream functions for file reading and writing in this project: fopen, fread, fseek, fprintf, fclose (consider ftell for debugging). Each of these functions needs a "FILE *" pointer as input. <u>Your program will be checked for good programming practices.</u> (Close your file streams, use memory correctly, variable initialization, etc.)

Also, add support for command line arguments (argc and argv) in the main function.

Your program should run as:

```
./proj2B <input_name> <output_name>
```

(The input_name will be 2B_binary_file, unless you change it.)

Finally, note that I am handing you a binary file. I think we are all using little endian systems, and so it will be fine. (It has been the last four years.) But, if it is big endian, then we will have a problem. You can verify you have little endian by printing the first byte (char) of the integer value 1 (don't submit this code):

```
unsigned int testInt = 1;
char *c = (char*)&testInt;
if (*c) printf("Little endian\n");
else printf("Big endian\n");
```

While testing your code and prior to submitting, run your source code file (.c) through the provided "grader.sh" shell script. It accepts your source file as an input argument:

```
./grader.sh <.c source file>
```

Take a look at the contents of the grader script. It contains initial tests that ensure you are following the specifications of this prompt. If you pass all tests, that only assures that your code compiles properly with the correct input and output. However, your actual source code will still be graded for good programming practices (as mentioned above).

Please submit only your source code (.c file).

We will be testing your code on ix-dev and use the following steps:

```
gcc –o proj2B proj2B.c
./proj2B /path/to/2B_binary_file /path/to/output
```

You should absolutely be testing your code on ix-dev. If it doesn't work there, you will be penalized heavily. And use the grader.sh script!