# CS 2275 Exercise 6 –Text & Binary Files

Module 6 Objectives: Read text files with get and put, read binary text files

Readings for Module 6 : Lippman - 7

Using Dev-C++, create a .cpp source file titled ex6-<your last name>.cpp that contains solutions to the following problems. As with all assignments in this course, be sure to include a block comment just about the function indicating its purpose and use a really good function name. Include code under main to test your function by allowing the user to enter the needed parameters. Make sure you test your function sufficiently to insure correctness

1. Using the code from note 6.2, write a function that uses the **put** function to write the following lines to a text file named output.txt:

|  |  |
| --- | --- |
| Line one |  |
| Line two |  |
| Line three |  |
|  |
|  |
|  |
|  |

1. Write a function that uses the **get(), fail(), and eof()** functions to read in the file generated from the code in problem 1.
2. Write a function that uses the input method to read in the file generated in problem 1.
3. Write a function writeUnLongLongToBinary, similar to readUnLongLongFromBinary in note 6.1 that writes out an array of unsigned long longs to a binary file. Test it using the provided code in note 6.1 and an array containing the integers 1 through 100.
4. Write a function that accepts a textfile name and a binary file name, reads the text file into a string, moves each asci value into an array of unsigned long longs, writes that array to a binary file. Write a second function with the same two arguments that reads the binary file into an array of unsigned long longs, recreates the string from the asci values, and writes out the string.Test your code using Persuasion (<http://www.gutenberg.org/cache/epub/105/pg105.txt> )

Grading rubric: Problems 1 through 5 : 20 points. Style up to -10: poor variable or poor function names or no block comment with a goal statement for a function. Solutions that have significantly more lines that needed will be docked points for lack of elegance.

**REMINDERS**

* code that does not compile will not be graded
* the grader should not need to modify/uncomment your code to test it. Provide a test mechanism allowing the grader to enter various tests for teach function.