CSCD 240

1) Create a C file name **txtToBinary.c**

   a) Opens an input text file (using fopen) by prompting the user for the filename. If the file does not exist or can't be opened the user is reprompted. If the file can't be opened the program will re-prompt. You may assume all files exist in the current working directory. Name your input file myTextInput.txt

   b) Opens an output file in binary mode (using the open system call) hard code the output filename to myBinary.bin. If the file can't be opened the program will display an error message and exit. You may assume all files exist in the current working directory. NOTE O_CREAT|O_WRONLY

   c) Reads a string from the input file (using fscanf) and writes the number of characters in the string and that string, in binary, to the output file (using the write system call). This will continue until the end of file. The program should exit if any write cannot be performed.

2) Create a C file name **binaryToTxt.c**

   a) Open the binary file myBinary.bin (using the open system call) whose name is hardcoded. If the file cannot be opened display an error message and exit.

   b) Write a function called **convertToText** that uses the read system call, to read from the binary file, and fprintf to write the original strings to the screen. This will continue until end of file is reached. The program should exit if any read cannot be performed.

   c) Write a function called **convert** that uses the read system call, to read from the binary file, and the write system call to write to the screen. This will continue until end of file is reached. The program should exit if any read or write cannot be performed. You will need to reset the file descriptor using the lseek system call at the top of this function.

NOTES

- You may fopen, fscanf, and feof, to open and read from the text input file.
- The commands for the binary file must be low level system call commands. You are writing binary files so don't write a carriage return.
- You will need to create your own files
- No line will be more than 100 characters so define your constant.
- The output to the screen should be the exact same as the input file.

TO TURN IN:

A zip file containing:

- myTextInput.txt
- myBinary.bin
- binaryToTxt.c
- txtToBinary.c

Your zip will be named: your last name first letter of your first name lab13.zip
(Example: steinerslab13.zip)