```
/***************************
Programer: Michael Marelli
Class: CS460
Project: Final
Date: 12/9/2016
File: l2u.c
***************************/
#include "ucode.c"

main(int argc, char *argv[])
{
    int in, out;
    int bread, i = 0, j = 0;
    char buf[1024], c, *tty; //c is used as input char

    gettty(tty);
    out = open(tty, O_WRONLY); //write to stdout
    in = 0; //use the stdin uless a file is passed in

    //this is adding extra characters when doing cat file > file
    if(argc >= 2) //file(s) provided to us
    {
        for(j = 1; j < argc; j++)
        {
            in = open(argv[j], O_RDONLY);
            while(bread = read(in, buf, 1024))
            {
                for(i = 0; i < bread; i++)
                {
                    if(buf[i] >= 'a' && buf[i] <= 'z')
                    {
                        buf[i] -= 32;
                    }
                    write(1, buf + i, 1);
                    if(buf[i] == '\n')
                    {
                        write(out, "\r", 1);
                    }
                }
            }
            close(in);
        }
    }
    //when output is redirected we still need to write to the tty
    else //use stdin as file
    {
        while(read(0, &c, 1))
        {
            if(c >= 'a' && c <= 'z')
            {
                c -= 32;
            }
            if(c == '\r') //clear the buffer to the output if input is stdin we need to print the buffer
            {
                write(out, "\r", 1);
                c = '\n';
                buf[i] = c;
                write(1, &c, 1);
                write(out, buf, strlen(buf));
                write(out, "\r", 1);
```

```
                bzero(buf, 1024);
                i = 0;
            }
            else if(c == '\n')
            {
                write(out, "\r", 1);
                buf[i] = c;
                write(1, &c, 1);
                //write(out, buf, strlen(buf));
                bzero(buf, 1024);
                i = 0;
            }
            else
            {
                buf[i++] = c;
                write(1, &c, 1);
            }
        }
    }
    close(out);
    exit(1);
}
```