

CSCI 1120 (Low-Level Computing), Spring 2010

Homework 5

Credit: 10 points.

1 Reading

Be sure you have read, or at least skimmed, the readings for 3/29, linked from the [“Lecture topics and assignments”](#) page¹.

2 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source by sending mail to `bmassing@cs.trinity.edu`, with each file as an attachment. Please use a subject line that mentions the course number and the assignment (e.g., “csci 1120 homework 5”). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department’s Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (10 points) In the early days of the Internet, a popular way of disguising text that some might find offensive was to encode it using a scheme called rot13 (short for “rotate 13”). In this scheme, all letters are rotated 13 positions (so ‘a’ becomes ‘n’, ‘b’ becomes ‘o’, ‘n’ becomes ‘a’, etc.). Spaces, digits, punctuation, etc., are not changed. (An advantage of this scheme is that if you apply it twice, you get the original text back. Think about why!)

Write a C program that, given two command-line arguments *infile* and *outfile*, encodes the text from *infile* using rot13 and writes the result to *outfile*. It should print an appropriate error message if called with fewer than two arguments, or if either file cannot be opened.

For example, if *infile* contains

```
Now is the time for all good persons to come to the aid of their
party. Hello world! 1234 !@#$
```

then *outfile* will contain the following

```
Abj vf gur gvzr sbe nyy tbbq crefbaf gb pbzr gb gur nvq bs gurve
cneql. Uryyb jbeyq! 1234 !@#$
```

Hints:

- Here is a partial function to do the required encoding.

¹http://www.cs.trinity.edu/~bmassing/Classes/CS1120_2010spring/HTML/schedule.html

```
int encode(int input_char) {
    if (('a' <= input_char) && (input_char <= 'm')) {
        return input_char + 13;
    }
    else if (('n' <= input_char) && (input_char <= 'z')) {
        return input_char - 13;
    }
    /* add code for uppercase letters, other characters */
}
```