— THE END —

Below are a few problems you should work out for yourself while you are studying for the final examination. We will also spend this lecture working on these exercises, and answering any questions you might have.

• **Problem 1: Comparing Strings**

Write your own version of the standard C library function `strcmp`, which compares two strings. The prototype of the function should be as below:

```
int strcmp(const char *a, const char *b);
```

• **Problem 2: Prime Factorization**

Write a C program that asks a user for an integer, and prints out its prime factors on the screen. For example, if the user enters 104, the result should be:

```
2 2 2 13
```

If the number is itself prime, it should print just the number itself back on the screen.

• **Problem 3: Scramble a Word**

Write a C function that takes a string (which is assumed to contain a word) as an argument, and scrambles the order of the letters in random fashion. For example, if the user enters “physics”, the result could be “cssipyh”. The prototype of the function should be as follows:

```
void scramble(char *word);
```

• **Problem 4: Cubes of Digits**

The number 153 has the interesting property that it is equal to the sum of the cubes of its digits:

$$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$

Write a program to find all three digit numbers that posess this property and print them out.

• **Problem 5: Count the Words**

Write a C program that counts the number of words in a text file. Note that in a proper text file, words are separated either by spaces or the end of a line. Punctuation never exists alone, and can be considered to be part of the word before it.