

# PERFORMANCE, DATA STRUCTURES AND ALGORITHMS

Exercise 10



Exercise 10

Read and write a unumber

# Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PURPOSE

The purpose of this exercise is to give you practice at working with structures in C.

Characters in C are encoded using the ASCII table (refer back to chapter 2 in your textbook). You can search the Internet to find the entire table. There are some useful characteristics of the ASCII table. One is that alphanumeric characters are listed in numerical order. Thus, the character ‘0’ is less than ‘1’. In fact, ‘0’ is exactly one less than ‘1’, since the ASCII code for ‘0’ is 48 and the ASCII code for ‘1’ is 49. This allows us to compare characters without converting them to numbers. It also allows an easy way to translate ASCII characters to numeric values. For example, the string “123” is comprised of 4 bytes with the values 49, 50, 51, and 0 (0 is the terminating 0 byte). Conveniently, we can easily translate this into a numeric byte array by subtracting 48 (the ASCII code for ‘0’) from each digit: 49-48, 50-48, 51-48 yields (numeric) 1, 2, and 3. Conversely, if you have a byte array with numeric values, you can add 48 to each value to get the printable ASCII character for that value.

For this exercise you are given a program that reads data from “stdin” (from the terminal). Your task in this exercise is to write a function to populate an unlimited precision number structure from the input values and a second function to print the contents of the structure.

The program repeatedly prompts the user for input until the user enters an empty line (hits return at the prompt). The input has three values separated by commas: <sign>, <decimalpower>, and digits”. <sign> is either + or -, decimalpower is an integer; digits are the digits in the number. All values are required (i.e. the sign is not optional).

Example input: +,1,123

Example output:

sign = positive, decimal power = 1, size = 3, value = 123

You entered the number 1.23

Example input: -,2,123

Example output:

sign = negative, decimal power = 2, size = 3, value = 123

You entered the number -12.3

You will not need to change main() for this exercise.

ACTIVITIES

Perform each of the following activities. If you have questions, issues, or doubts, please ask for help and do not just guess.

1. Complete the body of the new\_unumber\_from\_string() function so that it correctly stores the different parameter values into the UNumber parameter.
2. Add code to the body of the print\_unum\_struct() function so that the function displays a Unumber structure in the format of the *first* line of output in the examples above.
3. Test your program thoroughly with different values to verify that it works correctly.
4. Document your code carefully.
5. In your Engineering Notebook, explain your approach to creating your code and your testing strategy. Include examples of the test input you used and the corresponding output from your program.
6. When you are ready to submit your work, first remove all intermediate files from your src directory.
7. Save and archive your finished C program and upload it to the LMS.