

## Homework 4

**Due:** June 8, 2023

**Points:** 100

1. (35 points) The file “atomic\_weights.txt” contains lines with three fields separated by tabs. The first field is the atomic weight, the second field is the symbol for the element, and the third field is the name of the element . Write a program to load the contents of the file into an array, and then print out the array. The program is to take 1 argument on the command line, which is the name of the file. When you print it, print the weight as a floating point number and the symbol and name as strings. The three fields are separated by tabs. You can use this string as the format for printing: “%6.2f\t%s\t%s\n”. If you encounter one of the following errors, here’s what you print:
  - (a) There is more than 1 argument (that is, `argc` does *not* equal 2): print the following, where `%s` is replaced by the program’s name (`argv[0]`); “Usage: %s file\n”. Then return an exit status code of 1.
  - (b) The file cannot be opened: use  `perror` with the file name as an argument. Then return an exit status code of 1.
  - (c) The atomic weight is out of range (that is, larger than the maximum integer or smaller than the minimum number), print the following, with “%d” replaced by the line number: “Line %d: bad atomic weight\n” and skip the rest of the line. *Hint:* : Check the manual page for  `strtod(3)`.
  - (d) The line does not begin with an atomic weight: print the following, with “%d” replaced by the line number: “Line %d: missing atomic weight\n” and skip the rest of the line.
  - (e) If the line ends with only an atomic weight, and there is no symbol or name, print the following, with “%d” replaced by the line number: “Line %d: no element symbol or name\n” and skip the rest of the line.
  - (f) If the atomic symbol is one character, it must be capitalized; if it is 2, the first letter must be capitalized and the second lower case. If this is not so, print the following, with “%d” replaced by the line number and “%s” with the symbol on the line: “Line %d: bad atomic symbol: %s\n” and skip the rest of the line.
  - (g) If there is no name for the element, print the following, with “%d” replaced by the line number: “Line %d: missing element name\n” and skip the rest of the line.Call your program “atomic.c”. You can find an executable for it, and the file  `atomic_weights.txt`, in the directory  `/home/bishop/ecs36a/hw4`.
2. (35 points) Now modify the program you did in problem 1 to sort the elements. You can sort them in 3 ways.
  - (a) If the option  `-n` is given, sort the elements by name and print them in that order.
  - (b) If the option  `-s` is given, sort the elements by symbol and print them in that order.
  - (c) If the option  `-w` is given, or no option is given, sort the elements by atomic weight and print them in that order.So your program must take 1 (the file name) or 2 (the option and the file name) arguments. Call your program “atomicsort.c”. You can find an executable for it, and the file  `atomic_weights.txt`, in the directory  `/home/bishop/ecs36a/hw4`.  
*Hint:* You can write your own sort routine, or use the library function  `qsort(3)`.
3. (30 points) (30 points) The program  `getbit.c` (available in  `/home/bishop/ecs36a/hw4/getbit.c`) reads in two numbers,  `n` and  `b`. It returns the  `b`-th bit of integer  `n`, where the smallest (rightmost) bit is bit number 0. Rather, it is supposed to. But it doesn’t work. Please debug it. Put the debugged program into a file called “getbit.c”. Turn it in to Gradescope; you can do this as many times as you like, up to the due date, and we will grade the last one you submit.