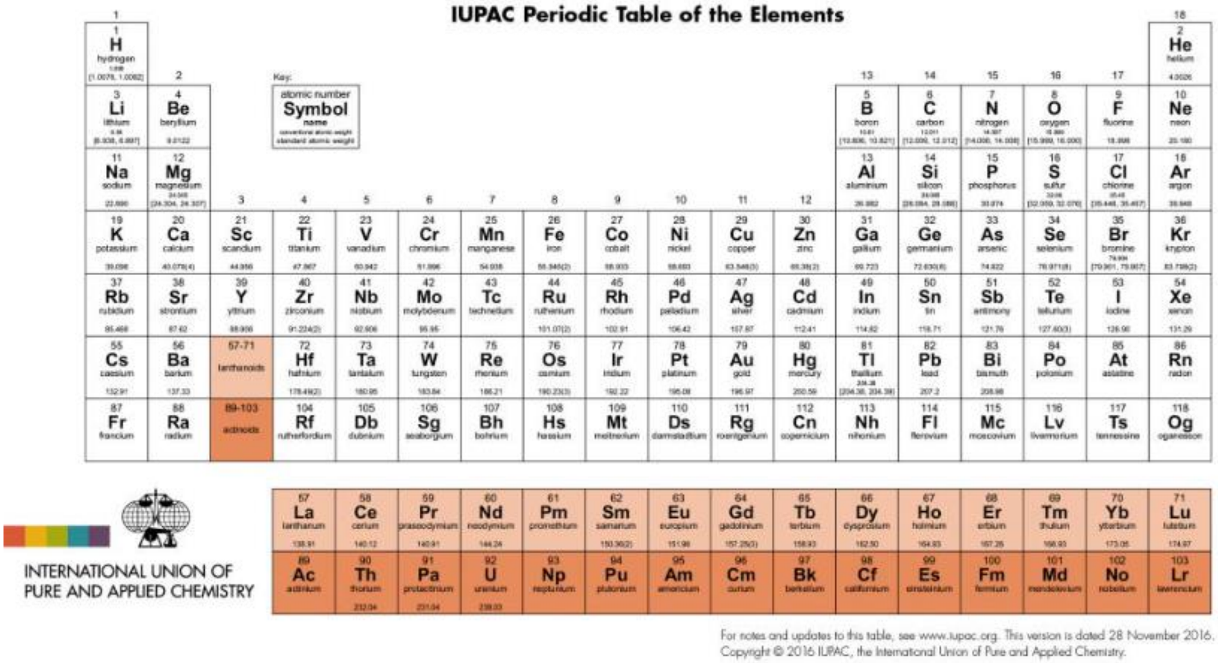
CS410P

ASSIGNMENT 10P

Scenario

The periodic table, or periodic table of elements, is a tabular arrangement of the chemical elements, ordered by their atomic number, electron configuration, and recurring chemical properties, whose structure shows periodic trends. Generally, within one row (period) the elements are metals to the left, and non-metals to the right, with the elements having similar chemical behaviors placed in the same column. Table rows are commonly called periods and columns are called groups. Six groups have accepted names as well as assigned numbers: for example, group 17 elements are the halogens; and group 18 are the noble gases. Also displayed are four simple rectangular areas or blocks associated with the filling of different atomic orbitals.

Each element in the periodic table has several properties but often we are not able to collect all their most commonly used properties together in one data set.


IUPAC Periodic Table of the Elements
INTERNATIONAL UNION OF PURE AND APPLIED CHEMISTRY
For notes and updates to this table, see www.iupac.org. This version is dated 28 November 2016.
Copyright © 2016 IUPAC, the International Union of Pure and Applied Chemistry.

Problem

Our data for each element in the periodic table may come from different sources. Let's take some of these characteristics of each element and see if we are able to put them together under one data structure so that we are able to make queries against them more easily.

We have two input files that we need to work with:

1) atomic_elements → input file containing atomic elements with number and weight
2) atomic_properties → input file containing melting point, boiling point and density of elements

The only common item that appears in both data files is the name of the atomic element. Our first task is to be able to set up a data structure from the first data set that contains the atomic element name, its abbreviation, its atomic number and atomic weight as four different items in one record. After we are done setting this up, we want to be able to add to the properties of these atomic elements with the additional information about melting, boiling points and density (if applicable) for each atomic element.

Once our data structure is completely set up with all the data we should be able to perform queries by specifying an atomic number and coming up with the details of that atomic element. Input is accepted continuously (integer values) to search by atomic number until the value 0 is entered by the user.

When you load the second data file you will notice that several values for melting point, boiling and density have the value -9999. This simply means that the particular property for that atomic element is either not applicable or unknown. While it's ok for you to load those values into your array, when you report on it when you query individual elements you must print those values as "Unknown"

Details

Your program must consist of two files, one containing the main() function and another containing all the other functions. Here's a list of suggested functions:

```
read_periodic_table()
print_periodic_table()
update_periodic_table()
search_periodic_table()
```

The output must first print out data stored in the data structure after reading the data from both files. As the user repeatedly searches for an atomic element by specifying the atomic number, the matching record details (or a message indicating record was not found) must be printed. Your output does not have to look exactly like mine but it should be as complete and easy to read. Make sure you indicate "Unknown" values when you encounter the value -9999 for any property for a specific atomic element. The program stops when the user enters 0 to stop searching.