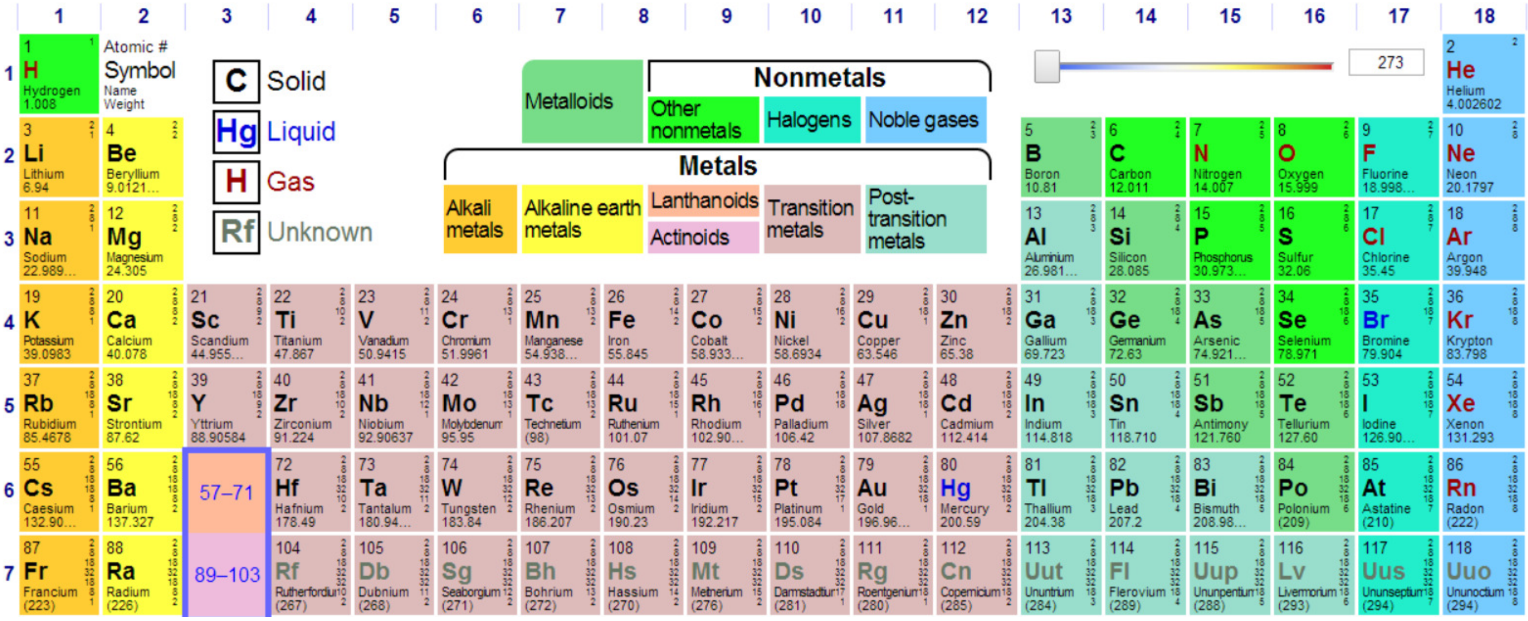
Lab 6 – this is your second graded lab… posted 10/19, and due SATURDAY Nov. 1 at 11:59:00 pm.

Working with Functions and Arrays: Construct arrays for the first 8 elements on the periodic table and add the given properties. The properties needed to be added are atomic number, name, symbol, and atomic mass to the thousandths. Then, using the program, show that you were able to edit the database by correcting the mismatched properties.



Part 1: Create the arrays: Two string arrays named Element Name and Symbol, one integer array named Atomic Number, and one double array named Atomic Mass. The arrays are created with [8] to hold 8 elements, which are indexed from 0 -7. There is a way to make the index display from 1-8 for easier understanding though, so look into that possibility.

First, the arrays must be defined outside of main so that they then will be accessible to main and then all the functions incorporating them.

It should look like this:

```
string elementName[8], elementSymbol[8];  // creates 4 empty arrays
int atomicNumber[8];
double atomicMass[8];
```

You have to preload the fields within main like so:

```
elementName[0] = "Hydrogen";
elementSymbol[0] = "C";
atomicNumber[0] = 1;
atomicMass[0] = 1.008;

elementName[1] = "Lithium";
elementSymbol[1] = "He";
atomicNumber[1] = 2;
```

```
atomicMass[1] = 4.003;

elementName[2] = "Helium";
elementSymbol[2] = "Li";
atomicNumber[2] = 3;
atomicMass[2] = 6.940;

elementName[3] = "Beryllium";
elementSymbol[3] = "Be";
atomicNumber[3] = 6;
atomicMass[3] = 9.012;

elementName[4] = "Oxygen";
elementSymbol[4] = "B";
atomicNumber[4] = 5;
atomicMass[4] = 10.810;

elementName[5] = "Carbon";
elementSymbol[5] = "H";
atomicNumber[5] = 4;
atomicMass[5] = 12.010;

elementName[6] = "Nitrogen";
elementSymbol[6] = "N";
atomicNumber[6] = 7;
atomicMass[6] = 16.000;

elementName[7] = "Boron";
elementSymbol[7] = "O";
atomicNumber[7] = 8;
atomicMass[7] = 14.010;
```

Don’t forget to #include <string>

Part 2: Create a menu that is easy to understand and that re displays itself after each operation.

Part 3: Allow the user to edit any property for the periodic table database. First, the user must be asked through the menu which property they would like to change.

Part 4: Use the program to edit the periodic table and correct any errors. Save an image of the screen after you utilized the program to edit the periodic table database and made sure each value is correct. The preload fields will still have the wrong values but the database you print after making all the changes should display the first 8 elements in the periodic table perfectly.

Hint 1: Read the lecture notes on Arrays and look at the sample code Arrays.cpp and myTunes.cpp. Working with Arrays can be a little difficult.

Hint 2: You must use functions to make the program easier to read and write. You can create your own functions, but these are suitable ones that you can use.

```
int printMenu( );
void printPeriodicTable( );
void changeElementName( );
void changeElementSymbol( );
void changeAtomicNumber( );
void changeAtomicMass();
```

Hint 3: Define your arrays OUTSIDE OF MAIN, so that they will be accessible to main and all of your functions.

Hint 4: a menu function you canuse (must be called from main).

```
int printMenu()
{
int choice = 0;
cout << "..............................." << endl;
cout << " 0 - exit                      " << endl;
cout << " 1 - print periodic table      " << endl;
cout << " 2 - change element name       " << endl;
cout << " 3 - change element symbol     " << endl;
cout << " 4 - change atomic number      " << endl;
cout << " 5 - change atomic mass        " << endl;
cout << "..............................." << endl;
while (true)
   {
   cout << "Enter choice: " ;
   cin >> choice;
   cout << endl;
   if ((choice >= 0) && (choice <= 5))
      {
      break;
      }
   else
      {
      cout << "Please enter a choice between 0 and 5." << endl;
      continue;
      }
   }//end while
return choice;
}//end printMenu
```

Hint 5: If you use the above function, you should call it from main with an assignment statement, and then a simple series of if…else statements can use the result to call the next correct function. You must handle incorrect inputs (that is, menu choices outside of the menu range).

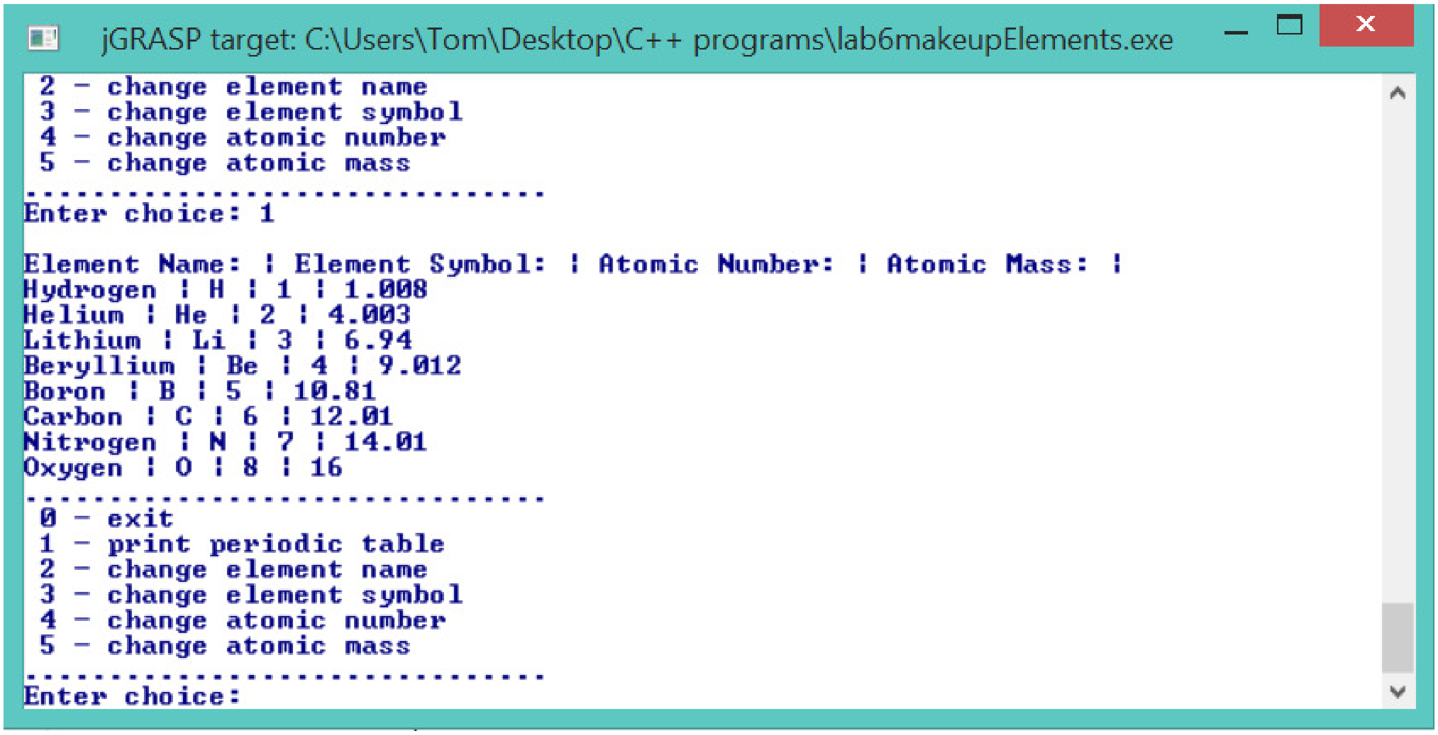
Hint 6: All of the change functions will have to first ask the user which element number to change. PLEASE NOTE: the name “index” CANNOT be used as a variable name anywhere in the program. “index” is a reserved C++ word. You must handle incorrect inputs (that is, array index values greater than 7 or less than 0).

Hint 7: Printing the database should display the new changed records instead of the original ones. Once you stop and restart the program, however, the changes will be lost.

Hint 8: you must handle incorrect inputs (that is, menu choices outside of the menu range, or array index values greater than 7 or less than 0).

Submit your cpp file to UB Learns. DO NOT submit your exe file. This lab will be graded.

Solution:


jGRASP target: C:\Users\Tom\Desktop\C++ programs\lab6makeupElements.exe

```
 2 - change element name
 3 - change element symbol
 4 - change atomic number
 5 - change atomic mass
-------------------------------
Enter choice: 1

Element Name: ¦ Element Symbol: ¦ Atomic Number: ¦ Atomic Mass: ¦
Hydrogen ¦ H ¦ 1 ¦ 1.008
Helium ¦ He ¦ 2 ¦ 4.003
Lithium ¦ Li ¦ 3 ¦ 6.94
Beryllium ¦ Be ¦ 4 ¦ 9.012
Boron ¦ B ¦ 5 ¦ 10.81
Carbon ¦ C ¦ 6 ¦ 12.01
Nitrogen ¦ N ¦ 7 ¦ 14.01
Oxygen ¦ O ¦ 8 ¦ 16
-------------------------------
 0 - exit
 1 - print periodic table
 2 - change element name
 3 - change element symbol
 4 - change atomic number
 5 - change atomic mass
-------------------------------
Enter choice:
```