Finding Molecular Weights

For each of the basic chemical elements, e.g. carbon, hydrogen, mercury, and gold, chemists have created a symbol comprised of one or two letters, in these cases C, H, Hg, and Au, respectively. The first letter is always upper case and the second, if there is one, is lower case. Each element has a certain atomic weight.

A common problem is to calculate the molecular weight of some compound, given the atomic weights of the elements which comprise the molecule. A table of atomic weights is standard and can be found in any chemistry textbook -- here are a few examples:

H	1.0079	O	15.999	F	18.9984
He	4.0026	Na	22.9898	Au	196.9665
Be	9.0122	S	32.06	Hg	200.5
C	12.011	Cl	35.453	Ra	226.0254

Some examples of chemical formula for some common molecules are:

Salt	NaCl
Water	H_2O
Sulfuric Acid	H_2SO_4

The subscript after an element symbol means that there are that many atoms of the corresponding element in the molecule. If no subscript is given, it means that there is only one atom of that element. If the element has a two character symbol, the first character is always upper case, and the second is always lower case. Since plain ASCII text contains no subscripts, chemical formula can be expressed with integers in the place of subscripts, e.g.

```
NaCl     H2O     H2SO4
```

Write a program which reads chemical formulas (in ASCII) and computes and displays the molecular weights. The program does not have to input the atomic weight table, the table may be declared and initialized internally to the program (i.e., "hard coded").

Assume subscripts in range 2-9.
Assume input is valid.

Time Recording Log

Name: Dalby
LOC Start:

Project / Module: Molecular Weights
LOC End: 27 logical stmts
64 physical lines

Date	Start	Stop	Interruption Time	Delta Time	Activity	Comments
10/20	1022	1059		37	DESIGN	DESIGN
10/21	1031	1051	1	20	CODE	on paper
	1125	1132		7	CODE	typing
10/22	1540	1606		26	REVIEW	trace
	1615	1616		1	COMP	No Syntax Errors
	1616	1622		6	TEST	Create unit tests w/ BlueJ

Total: 97

Molecular Weights Program

Test Data

H	1.0
He	2.0
C	3.0
O	4.0
Na	5.0
Cl	5.5
P	6.0

Input	expected output
H	1.0
H2	2.0
He	2.0
He2	4.0
HC	4.0
O2Cl	~~15~~ 13.5
C3O	13.0
C3PO	19.0
NaP	11.0
CH9	12.0
HeCl	7.5
Na2He2	14.0

Algorithm Design

Maybe append a sentinel so peek() won't crash.

e.g. Na2He2 ← Are we at end?
↑ ← Is next symbol a digit? lowercase
↑↑ ← Is second ~~letter~~ symbol a letter?

First letter will always be upper case.

Use lookup String? or array?

"H He C O Na Cl P " and parallel weight array.

then use indexOf(2-char element)

↑ Pad 1-char element with trailing blank

Each entry is 2-char.

```
Total = 0
Loop through entire expression
    Multiplier = 1
    Element = current char
    IF peek is lowercase letter
        Append it to element
        Advance current
oops→ ELSE Append blank
    IF peek is digit
        multiplier = digit
        Advance current
    Total += lookupElementWeight() * Multiplier
End loop
Return Total
```

← oops. Advance current

Algorithm Trace

Input 'H'

Total	current	mult	element
0	0	1	H
1.0			

Input 'He'

0	0	1	H
	1		He
2.0			

Input 'Hc'

0	0	1	H
1.0	oops! current didn't advance		
0	0	1	H
1.0	1	1	C
4.0			

Input 'HeC'

0	0	1	H	oops, didn't append blank
	1		He	to single-char element
2.0	2	1	C_	
5.0				

Input 'O2P'

0	0	1	O_
	1	2	
8.0	2	1	P_
14.0			

```
/** Find Molecular Weights */

public class Molecules
{
                        String
   private final static symbolLookup
      = "H HeC O NaClP ";
   private final static double[] weightLookup
      = { 1.0, 2.0, 3.0, 4.0, 5.0, 5.5, 6.0 };

   public static double findMolecularWeight (String molecule)
   {
      // Append a sentinel character to string
      molecule = molecule + '.';
      double total = 0.0; // total weight

      int current = 0;    // current position in string
      // Loop through all symbols in molecule
      while ( molecule.charAt(current) != '.')
      {
         int multiplier = 1;    // default multiplier
         String element = molecule.charAt (current);

         char peek = molecule.charAt(current + 1);

         if ( Character.isLowerCase(peek))
         {
            element += peek;
            current ++;
         }
         else
         {
            element += ' ';
         }
```

```
            peek = molecule.charAt(current + 1);
            if ( Character.isDigit(peek))
            {
                multiplier = (int)peek - (int)'0';
                current ++;
            }

            // Do lookup in tables
            int offset = symbolLookup.indexOf(element);
            float weight = weightLookup[offset/2];

            total += weight * multiplier;
            current ++;
        }
        return total;
    }
}
```

```java
/**
 * Find Molecular Weights.
 *
 * @author J. Dalbey
 * @version 10/2009
 */
public class Molecules
{
    // Lookup table for element symbols
    private final static String symbolLookup = "H HeC O NaClP ";
    // Lookup table for element weights
    private final static double[] weightLookup =
    {
        1.0, 2.0, 3.0, 4.0, 5.0, 5.5, 6.0
    };
    // sentinel character to mark end of formula
    private final static char kSentinel = '.';

    /** Find weight of a molecule.
     * @param molecule the string containing the chemical formula
     * @return double the weight of the molecule
     */
    public static double findMolecularWeight(String molecule)
    {
        // Append a sentinel character to formula
        molecule += kSentinel;
        double total = 0.0;  // total weight
        int current = 0;  // current position in formula

        // Loop through all symbols in formula
        while (molecule.charAt(current) != kSentinel)
        {
            int multiplier = 1;  // default multiplier
            String element = "" + molecule.charAt(current);

            char peek = molecule.charAt(current+1);
            // Is it a two-char element?
            if (Character.isLowerCase(peek))
            {
                element += peek;
                current++;
            }
            else
            {
                element += ' ';
            }

            peek = molecule.charAt(current+1);
            // Does it have a subscript?
            if (Character.isDigit(peek))
            {
                multiplier = (int) peek - (int) '0';
                current++;
            }

            // Go lookup element in table
            int offset = symbolLookup.indexOf(element);
            double weight = weightLookup[offset/2];
            total += weight * multiplier;
            current++;
        }
        return total;
```

is arg a String? ✓

double ~~float~~

molecule = 'H'

" = "H."

total	current	multipler	element	peek	offset	weight
0.0	0	1	H	.		
45			H_			
48				.		
57					0	
58						1.0
59	1.0					
60		1				
62	1.0 ✓					

molecule = "HZ"

= "HZ."

0.0	0	1	H	2	
			H_		
				2	
52		2			
53	1				
57					0
58					1.0
59	2.0				
60	2				
62	2.0 ✓				

26 molecule = "HeZ."

0.0	0	1	H	e	
40			He		
	1				
48				2	
		2			
	2				
57					2
					2.0
59	4.0				
	3				

```
molecule = "C3PO."
0.0    0    1    C    3
                 C_   3
          3
     1                        4
                                   3.0
9.0
     2
          1
                 P    O
                 P_   O
                           12
                                6.0
15.0
     3
          1      O    .
                 O_   .
                           6
                                4.0
19.0
     4
19.0
```

molecule = "CH9."

0.0 0 1 C H

C_

H

4

3.0

3.0

1

1

H

9

H_

9

9

2

0

1.0

12.0

3

12.0 ✓

Defect Types		
10 Syntax	50 Architecture	90 Build, Package
20 Assignment	60 Data	100 Environment
30 Algorithm	70 Checking	110 System
40 Interface	80 Documentation	

Tally Marks by Inject Phase	
- Design	X Compile
/ Code	✇ Test
0 Code Review	

Defect Tally

Student DALBEY Date 10/22

Program Molecules Program # _____

Removal Phase _____

1 2 3 4 5 6 7 8 9 10

Type Description

ZERO DEFECTS!

Removal Phase _____

1 2 3 4 5 6 7 8 9 10

Type Description