2. PROGRAMMING ASSIGNMENT 2:
Read:        Textbook:  Chapter 5
Programming: Name your program p2.c
DUE:         Wednesday, August 16, 2017 @ 6:00am

Write a program to convert from kilograms to pounds and to stones (UK, IRE, AUS) using the while, for, and do loop. Use the formulas of “pound equals (kilogram * 2.2)” and “stone equals (pound * 14)”.

a) Use type char for “answer” to repeat program and double for variables to hold all weights and step factor.
   e.g. int count;  // Loop counter
        double step;  // Step factor to next weight
        double kg;   // Weight in kilogram
        double input_kg; // Input weight in kilogram

b) Use these symbolic constants for the conversion factors:
   #define KG_TO_LBS 2.2
   #define LB_TO_STONE 14

c) Allow the user to enter one positive real number, type double, for input_kg. The grader will input one number ONLY (no alphabetical characters).
   The kilogram and converted pounds and stones are displayed with 2 places of precision. e.g. printf ("%7.2lf %14.2lf %14.2lf\n", kg, lb, stone);

d) Use the while loop to generate a table of 5 of these calculations starting from the input kilogram weight and increasing with step factor.

e) Repeat this calculation using a for loop.

f) Repeat this calculation using a do loop.

g) Therefore your program will generate three identical tables for ONE kilogram input and ONE step factor by using while loop for the 1st set of calculations, for loop for the 2nd set of calculations, and the do loop for the 3rd set of calculations.
   ▶ Remember to reinitialize the starting “count” to one before each loop (while, for, do).
   ▶ Do NOT use "goto" in CSE 5a.

h) Leading zeros are ignored on input. Do not use functions, if-else, switch statements or any other construct we have not yet covered in lecture. Use only loop constructs of while, for, and do.

i) Comments are a very important part of your program. They tell the programmer what is being done and how it's being done. How do you know that your program is producing correct output? printf() in the program is very helpful to the user.
   printf() makes it easier to interact with your program making it user friendly.

j) Allow your program to repeat this calculation as often as the user wishes when the user types in a single character of ONLY a 'y' (lowercase y). Use the following code to handle character input:

```c
printf("Enter a positive real number in kilograms: ");
scanf("%lf", &input_kg);  // Read type double input
getchar();                // Read <ENTER> key
/* ... */
printf("Want to convert more Kilograms (y/n)? ");
answer = getchar();       // Assign to character
}while ( answer == 'y');  // Loop while a 'y' is input
```
**HINTS:** Code in this sequence.

Submit your program for grading as in Programming Assignment One (PA#1), only with a different number of "2".

a) Write code for user input, step a), b), c), and j). Write code for positive input later.

b) Modify **while** loop code step d) as given in lecture. Compile. Run. Test.

c) Write **for** loop code, step e). Test.

d) Write **do** loop code, step f) and g).

e) Write code to repeat calculations, step j). Write code for positive input adding an inner loop.

f) Use **ONLY** loop constructs and **NO** if-else or switch constructs. Step g) and h)

g) Write comments, step i). Done!

h) Test for ending program. Test.

Know the difference between integers and floating points. Which can store the larger (or smaller) number?

**Save the final version of your program as “p2.c”**

**PA#2 SAMPLE OUTPUT:** (input shown in **bold** type)

```
This program prints a table of weights in kg, lbs, and stones.

Enter a positive real number in kilograms: 1
Enter a positive real number step factor: 20

KILOGRAMS     POUNDS         STONES  (while loop)
==========     =======       ========
1.00           2.20          30.80
21.00          46.20         646.80
41.00          90.20         1262.80
61.00         134.20        1878.80
81.00         178.20        2494.80

KILOGRAMS     POUNDS         STONES  (for loop)
==========     =======       ========
1.00           2.20          30.80
21.00          46.20         646.80
41.00          90.20         1262.80
61.00         134.20        1878.80
81.00         178.20        2494.80

KILOGRAMS     POUNDS         STONES  (do loop)
==========     =======       ========
1.00           2.20          30.80
21.00          46.20         646.80
41.00          90.20         1262.80
61.00         134.20        1878.80
81.00         178.20        2494.80

Want to convert more Kilograms (y/n)? y
```

**PA#2 (continued next page)**
**PA#2 SAMPLE OUTPUT:** (input shown in **bold** type)

<table>
<thead>
<tr>
<th>KILOGRAMS</th>
<th>POUNDS</th>
<th>STONES</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5.55</td>
<td>12.21</td>
<td>170.94</td>
</tr>
<tr>
<td>11.10</td>
<td>24.42</td>
<td>341.88</td>
</tr>
<tr>
<td>16.65</td>
<td>36.63</td>
<td>512.82</td>
</tr>
<tr>
<td>22.20</td>
<td>48.84</td>
<td>683.76</td>
</tr>
</tbody>
</table>

**KILOGRAMS** | **POUNDS** | **STONES** (while loop)

- 0.00
- 5.55
- 11.10
- 16.65
- 22.20

**KILOGRAMS** | **POUNDS** | **STONES** (for loop)

- 0.00
- 5.55
- 11.10
- 16.65
- 22.20

**KILOGRAMS** | **POUNDS** | **STONES** (do loop)

- 0.00
- 5.55
- 11.10
- 16.65
- 22.20

Want to convert more Kilograms (y/n)? **N**

**Note:**
If you must interrupt your program (an infinite loop), type: `<CTRL>C` (^C)

Verify you saved your work in the **Documents - cs5v HOME directory**.