CS 150 Lab 4
Conditional Logic & Formatted Output

The main objective of today's lab is to solve a problem that uses conditional logic and formatted output. You are to continue using the Visual Studio debugger to help identify any logical errors that your program contains.

1. Be sure your output looks exactly like the specified output.
2. Submit your solutions folder with the completed project to **CS150-02 Drop** folder when you are done.
3. Use the program skeleton and add comments to your code.
4. Write small pieces of code and test as you go.

Show the instructor or TA the output of your menu before moving on. And then finally, show the instructor or TA your final solution.

Lab 4.1

Add a project called **04_1_SpeedOfSound** to your **PUNetIDLabs** solution that solves the following problem.

The speed of sound traveling through a gas medium depends on the density of the medium. The less dense the medium, the faster the speed of the sound travels. The following table shows the speed of sound at 0 degrees Celsius.

Consider the following table showing the speed of sound through a gas medium:

Medium	Speed (Meters/Second)
Carbon Dioxide	258.0
Air	331.5
Helium	972.0
Hydrogen	1,270.0

Write a program that displays a menu allowing the user the ability to select one of the four gases. The program is to ask the user to enter the time it took (in seconds) for the sound to travel from the source to the detection location. The program is to report how far away (in meters) the source of the sound is from the detection location.

1) Check to make sure the user has selected a valid menu choice. If an invalid choice is entered, output the message "Illegal Menu Choice" and terminate the program.

2) Optional Challenge: Do not accept any input times less than 0 or more than 60 seconds.

Here is how your program is to work:

```
C:\Windows\system32\cmd.exe                                    [ □ ] [ ▫ ] [ ✕ ]
12345678901234567890123456789012345678901234567890123456789012345678901234567 89
                         Speed of Sound in Gas

           Medium                      Speed (Meters Per Second)
           ------------------------------------------------------
           [C]arbon Dioxide                          258.00
           [A]ir                                     331.50
           [H]elium                                  972.00
           H[y]drogen                               1270.00

           Enter Medium: C

           Enter Seconds Sound Traveled: 10

           Distance of Sound from Detection Device: 2580.00 meters

Press any key to continue . . .
```

IMPORTANT: When outputting the menu, I do not want to see the following statement:

```
cout << "                      [C]arbon Dioxide                      258.0" << endl;
```

Instead, your program is to use constant variables with setw, fixed, and setprecision where necessary and be easily modifiable.

Note:

1. You must output the numbers at the top of the output. They can be used as a guide to ensure that your spacing is correct.

2. When you output anything with **setw**, then by default it will right align it. To align the output to the left, you will need to use the keyword **left**.

   ```
   cout << setw(5) << left << "oh" << setw(5) << right
       << "no" << endl;
   ```

   will output:

   ```
   oh      no
   ----------
   ```

3. Your program is to compile without any errors or warnings.

4. Do not use any magic constants in your program.

Once your project is complete, place your solution PUNetIDLabs into the
CS150-02 Drop folder on Turing. Your solution is to have all previous projects completely working and correct.

Optional Challenge:

The goal of the challenge is to write a program to determine if a year is a leap year or not.

A year is a leap year if it is divisible by 4. The only exception to this is if it is a century year. Then it is a leap year only if it is divisible by 400.

In case you're wondering, these are the rules for the Gregorian calendar, which began to be adopted in 1582 when they realized that having a leap year every 4 years resulted in the days of the year being very off after a several centuries. (The actual length of a year is 365.24219 days, not an even 365.25 days).

The program should accept a single year as input and output either "Leap year" or "Not a Leap year."

Questions:
- What years do you need to use to test your program to make sure that you've covered all possibilities?
- Can you determine a single relational expression that represents the above rules and use this in your program?