CSCI 1133
Exercise Set 2

You should attempt as many of these problems as you can. Practice is *essential* to learning and reinforcing computational problem solving skills. We encourage you to do these without any outside assistance. Create a directory named `exercise2` and save each of your problem solutions in this directory. Name your individual Python source files using "`ex2`" followed by the letter of the exercise, e.g., "`ex2a.py`", "`ex2b.py`", "`ex2c.py`", etc.

Note that we will provide automatic testing of your solutions using GitHub, so it is necessary to name your source files precisely as shown.

Recall from Lab3 that you need to include the following statement to the global namespace in order to automatically cause the `main()` function to be called when the module is loaded and executed:

```
if __name__ == '__main__':
    main()
```

A. **Basal Metabolic Rate Redux**

Write a well-structured Python program that will solicit a user to input his or her weight in pounds, height in inches, age in years, and the character 'M' for male or 'F' for female, then output the number of 230 calorie chocolate bars that need to be consumed to maintain their weight.

Start by converting the `bmr` algorithm you completed in exercise set 1 to a Python function, then create a non-pure `main()` function that will input the required information, call the `bmr` function and output the result.

B. **Baby It's Cold Outside**

Write a well-structured Python program that will do the following:

- Prompt the user and input two separate values for ambient temperature and wind velocity, respectively.
- Compute and output the wind-chill value

The formula for the wind-chill temperature, $t_{wc}$, is provided by the following formula,

$$t_{wc} = 35.74 + 0.6215t_a - 35.75v^{0.16} + 0.4275t_a v^{0.16}$$

where $t_a$ is the ambient temperature in degrees Fahrenheit and $v$ is the wind speed in miles per hour.

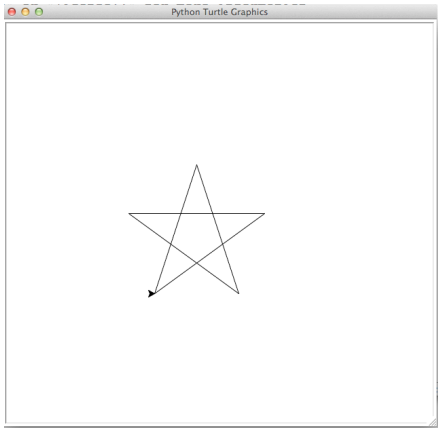
Use procedural decomposition and implement the wind-chill computation as a separate function. Call the wind-chill function from the `main()` function.

Examples:

```
>>> python3 ex2b.py
Enter temperature (F): 5
Enter wind velocity (MPH): 20
The windchill is: -15.435721635148337

>>> python3 ex2b.py
Enter temperature (F): -20
Enter wind velocity (MPH): 30
The windchill is: -53.02841296896488
```

C. **Turtle Star**
Write a well-structured Python program that draws a 5-pointed star using Python Turtle graphics:




Your program should do the following:

- Open a turtle dialog box and have the user enter the length of each side.
- Draw a 5-pointed star (as shown) with each side length as entered by the user. (Hint: the internal angle of each point is 36 degrees)

You should first solve the computational problem using pencil and paper pseudo code before translating it into Python!

D. **Equation of a Line**
Write a well structured Python program that prompts the user to enter the coordinates of two points $(x_1, y_1)$ and $(x_2, y_2)$, and outputs the slope-intercept equation of the line that connects the two points. Recall that the formula for the slope of the line is:

$$(y_2 - y_1) / (x_2 - x_1)$$

and the general slope-intercept form is:

$$y = mx + b$$

Using procedural decomposition, first construct/test a pure function named `slope` that will take $x_1$, $y_1$, $x_2$, and $y_2$ as arguments and return the slope of the line. After you have it working, construct the non-pure `main` function that will prompt the user for the points, call the `slope` function and output the equation of the line.

You should first solve the computational problem using pencil and paper pseudo code before translating it into Python!

Examples:

```
>>> python3 ex2d.py
Enter first x value: 1
Enter first y value: 1
Enter second x value: 4
Enter second y value: 4
y = 1.0 x + 0.0
```

```
>>> python3 ex2d.py
Enter first x value: 5.5
Enter first y value: 10
Enter second x value: 25.5
Enter second y value: -30
y = -2.0 x + 21.0
```

**Consider:** what happens if $x_1$ is the same as $x_2$? How would you fix your program to handle this?