ICA07 – Simulating Calories Burned

SCENARIO 1:

Running on a treadmill a runner burns 4.2 calories per minute. Write a program that uses a loop to display the number of calories burned after 10, 15, 20, 25, 30, etc. minutes.

Your function should take an input, a variable named "iterations", and it returns a string of comma separated values.
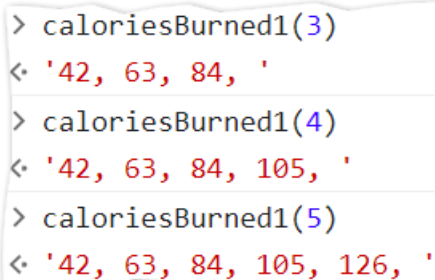
If the iterations parameter has the value of 5, then your function's output should look like this:
 42,63,84,105,126,

If the iterations parameter has the value of 6, then your function's output should look like this:
 42,63,84,105,126,147,

Notice that, in this scenario, it is OK to have a comma at the end of the output. Also notice that there is no HTML input form in this scenario. The focus is on completing the function caloriesBurned1. When you are done, you should be able to issue commands from the Chrome Web Developer console like this:

```
> caloriesBurned1(3)
<· '42, 63, 84, '
> caloriesBurned1(4)
<· '42, 63, 84, 105, '
> caloriesBurned1(5)
<· '42, 63, 84, 105, 126, '
```

INSTRUCTIONS:

1. Find a friend to work with. Retrieve the ica07_calories.zip provided by your instructor.

2. Extract the code into your mis2402workspace and open the treadmill1.html file in Visual Studio Code.

3. Write the code in the caloriesBurned1 function to get the output described above.

   HINT: It will be to your advantage to first figure out how to generate the sequence 10, 15, 20, 25, etc. After you do that, the problem gets a lot easier. There is more than one way to do this!

   ANOTHER HINT: Your goal here is to return a string. That means that you will need to have a variable that starts out as an empty string ('') and new values get concatenated to it with each step of the loop.

SCENARIO 2:

4. Keep working with your friend. Open the treadmill2.html file in Visual Studio Code.

5. Copy your code for the caloriesBurned1 function from treadmill1.html into the caloriesBurned2 function in treadmill2.html.

6. In this new scenario … it is *not* ok to end your output with a comma. Improve your original caloriesBurned1 code so that if the iteration is the last iteration, then you do not concatenate the "," on to the end of the string.

HINT: This means (of course) that you will need an if statement inside the loop. There are other ways to solve this problem, but the point here is that the code block of a loop may contain conditional statements … or even other loops.

7. Here's an example of what your testing should look like.

```
> caloriesBurned2(3)
<· '42, 63, 84'
> caloriesBurned2(4)
<· '42, 63, 84, 105'
> caloriesBurned2(5)
<· '42, 63, 84, 105, 126'
```

No comma!

SCENARIO 3:

Loops can also be used to generate repetitive sequences of HTML. So, in this scenario your caloriesBurned3 function will generate an unordered list in HTML instead of comma separated values.

If you don't remember the syntax for an unordered list in HTML, you will need to go look it up on W3Schools. https://www.w3schools.com/html//html_lists_unordered.asp

When you are done, you should be able to click the "Calculate" button and see output as shown below.

🔥 **Calories Burned**

Click the button to determine the calories burned over time.

[Calculate]

- 42
- 63
- 84
- 105
- 126
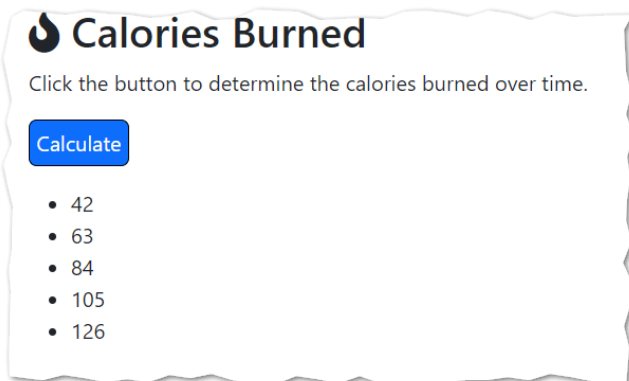
8. Keep working with your friend. Open the treadmill3.html file in Visual Studio Code.

9. Copy your code for the caloriesBurned1 function from treadmill1.html into calorieBurned3.

   HINT: You don't need a conditional statement to do complete scenario 3… that is why you are starting with the treadmill1 code again.

10. Edit your code so that the function returns an HTML unordered list.

11. As before, you should test your work using the Web Developer console. Below is an example:

```
> caloriesBurned3(3)
← '<ul><li>42</li><li>63</li><li>84</li></ul>'
> caloriesBurned3(2)
← '<ul><li>42</li><li>63</li></ul>'
> caloriesBurned3(1)
← '<ul><li>42</li></ul>'
```

Observe that, in the example above, the HTML sequence begins with <ul> and ends with .

Also notice that there are no line breaks in the HTML sequence. Your browser does not care about line breaks when evaluating HTML!

12. When you are done, upload all three treadmill files to ICA07 on canvas.

How will this activity be graded?

Your instructor will check your work for completion.

Task	Points
treadmill1.html	30
treadmill2.html	40
treadmill3.html	30
Total:	100