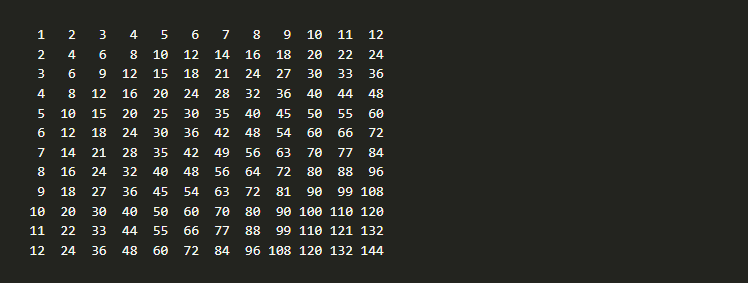
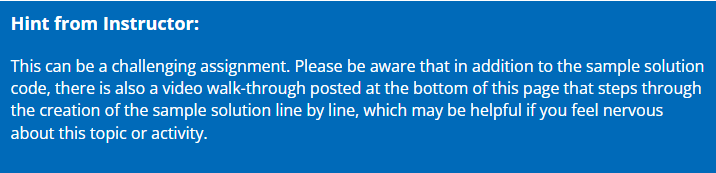
**Loops of Loops**

## Overview

When learning multiplication in school, students often write out multiplication tables where numbers are written along the top of each column and on the left side of each row. Then, the student fills in each spot in the table as the product of the number at top of the column and the left of the row. This might produce something like the following:

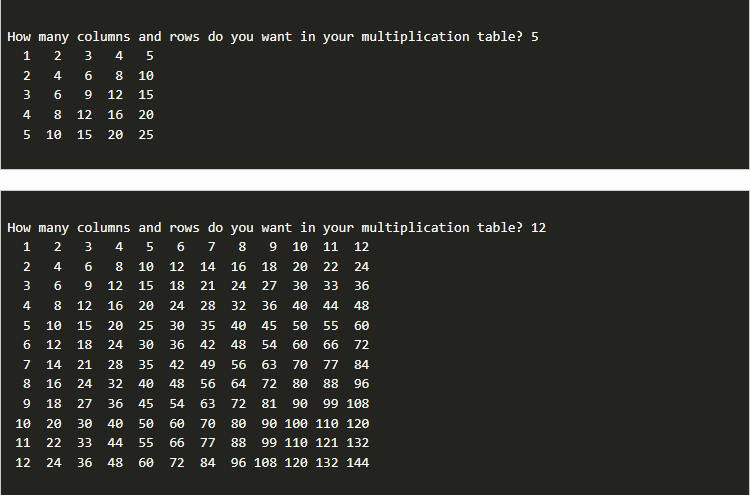
****

In this activity you'll write a program that uses loops within loops to generate multiplication tables.

****

## Assignment

Write a program that asks the user for the number of rows and columns in the table and then generates the associated multiplication table. The following are examples of the finished assignment:

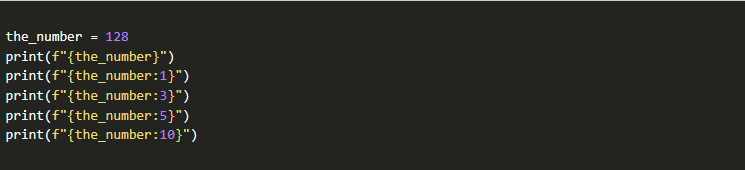
****

When complete, this program doesn't require too many lines of code, but they can be conceptually difficult to combine together when you are first learning to work with loops, so please work through the steps of this assignment, rather than writing the whole program at once.

To complete the assignment, you need a few other tools.

### PRINTING NUMBERS THAT TAKE UP A CERTAIN AMOUNT OF SPACE

In previous lessons you've used formatting strings to display numbers with a certain number of decimal places **"{the\_number:.2f}"**. In a similar way, you can specify the number of spaces to take up by including that number after the colon as follows:

****

This outputs the following:

****

Notice that if the number you specify is less than or equal to the number of digits in the number, it will simply display the number normally.

### PRINTING WITHOUT NEW LINES

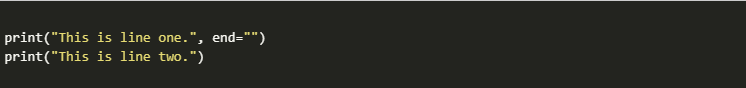
To this point, whenever we have used a print statement, it has always been on it's own line, so that the next line starts on a new line. If you do not want the print statement to end with a new line, you can specify the **end** as follows:

****

This outputs the following:

****

Along the same lines, if you do not want anything at the end, you can specify **end=""** as follows:

****

This outputs the following:

****

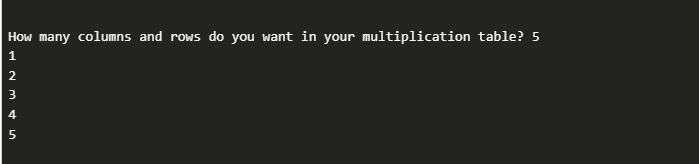
Notice, that because you told the first print statement to end with nothing (by using **""**), it does not end with a newline and the next line prints directly following it.

### CORE REQUIREMENTS

1. Ask the user for the number of rows and columns (note: you are only asking for a single number that will be used for both the rows and the columns).

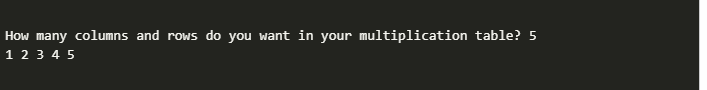
Display the numbers 1 up to and including that number, each on their own line at this point.

The output should look as follows:

****

1. Change the program so that the numbers are printed on the same line with a space between them.

The output should look as follows:



1. You now have code that can print out each column you'll need on a line. Now, you need to repeat it for a second line, and a third line, and so forth up until the number of rows that you need.

To accomplish this, add another loop for each row that has your previous code as its body. Start by simply having it duplicate the same row over and over again. Keep in mind that you'll need to display a newline after the numbers for each row are finished.

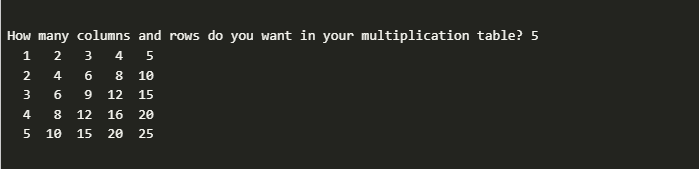
Then, once this is working change it to display the product of the row and column.

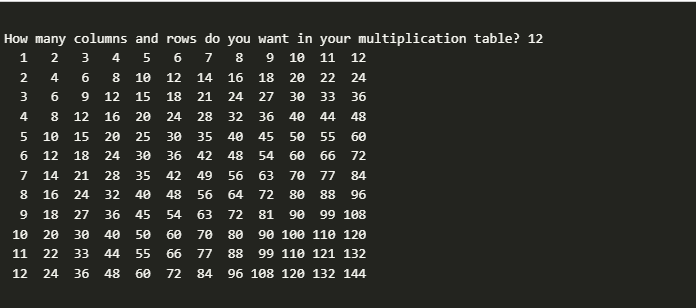
The output at the end of the core requirements should look as follows:



### STRETCH CHALLENGE

1. Change your program so that each number that is displayed takes up 3 spaces. This should look as follows:



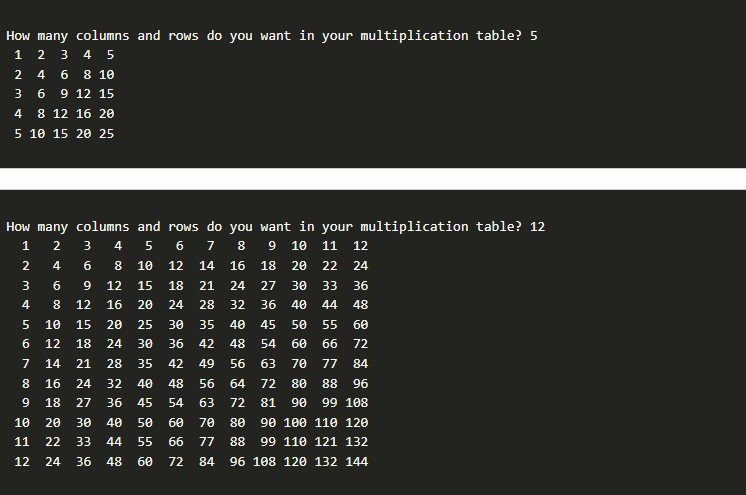


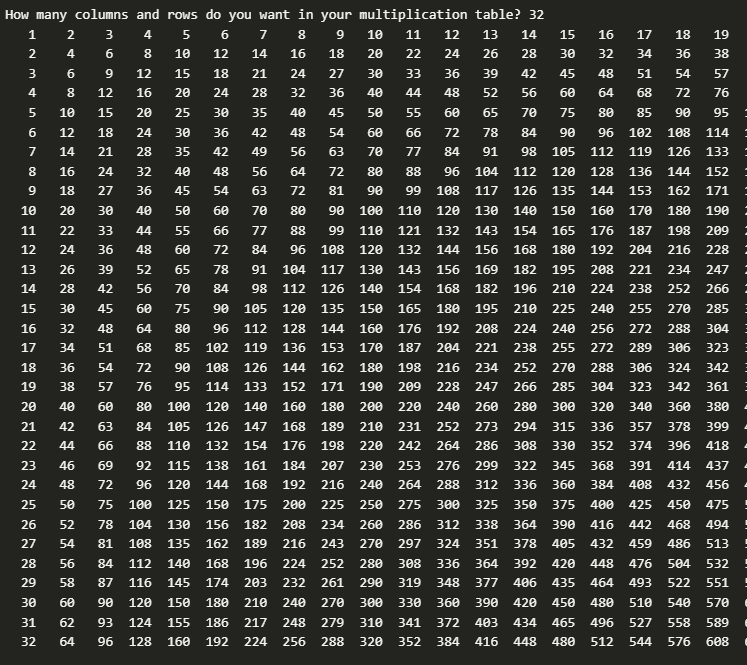
1. Notice that in the example of 5 rows and columns, you don't actually need three spaces per number, but instead two will do. Change your code so that it determines the number of spaces to use automatically, based on the largest number that will be produced.

For a first approach to this, consider using an if statement to see if the largest number would be greater than or equal to 100.

1. For a second approach to the problem of determining the number of spaces per number, consider mathematically determining the number of digits so that your table could work for any number of digits. (You can now do multiplication tables that go into the thousands and beyond!)

After the final stretch challenge you program can produce the following:





## Submission

When complete, please report your progress in the associated I-Learn quiz.

If you decided to do additional work on the program after your team activity, either by yourself or with others, feel free to include that additional work when you report on your progress in I-Learn.