

Sci...

Scientific Computing with Python Projects

Arithmetic Formatter ●

You will be working on this project with our Replit starter code.

Students in primary school often arrange arithmetic problems vertically to make them easier to solve. For example, "235 + 52" becomes:

```

  235
+  52
-----

```

Create a function that receives a list of strings that are arithmetic problems and returns the problems arranged vertically and side-by-side. The function should optionally take a second argument. When the second argument is set to `True`, the answers should be displayed.

Example

Function Call:

```
arithmetic_arranger(["32 + 698", "3801 - 2", "45 + 43", "123 + 49"])
```

Output:

```

  32    3801    45    123
+ 698  -   2  + 43  + 49
-----

```

Function Call:

```
arithmetic_arranger(["32 + 8", "1 - 3801", "9999 + 9999", "523 - 49"], True)
```

Output:

```

  32      1    9999    523
+  8  - 3801  + 9999  - 49
-----
 40  -3800  19998   474

```

Rules

The function will return the correct conversion if the supplied problems are properly formatted, otherwise, it will **return** a **string** that describes an error that is meaningful to the user.

- Situations that will return an error:
 - If there are **too many problems** supplied to the function. The limit is **five**, anything more will return: Error: Too many problems.
 - The appropriate operators the function will accept are **addition** and **subtraction**. Multiplication and division will return an error. Other operators not mentioned in this bullet point will not need to be tested. The error returned will be: Error: Operator must be '+' or '-'.

- Each number (operand) should only contain digits. Otherwise, the function will return: Error: Numbers must only contain digits.
- Each operand (aka number on each side of the operator) has a max of four digits in width. Otherwise, the error string returned will be: Error: Numbers cannot be more than four digits.
- If the user supplied the correct format of problems, the conversion you return will follow these rules:
 - There should be a single space between the operator and the longest of the two operands, the operator will be on the same line as the second operand, both operands will be in the same order as provided (the first will be the top one and the second will be the bottom).
 - Numbers should be right-aligned.
 - There should be four spaces between each problem.
 - There should be dashes at the bottom of each problem. The dashes should run along the entire length of each problem individually. (The example above shows what this should look like.)

Development

Write your code in `arithmetic_arranger.py`. For development, you can use `main.py` to test your `arithmetic_arranger()` function. Click the "run" button and `main.py` will run.

Testing

The unit tests for this project are in `test_module.py`. We are running the tests from `test_module.py` in `main.py` for your convenience. The tests will run automatically whenever you hit the "run" button. Alternatively you may run the tests by inputting `pytest` in the console.

Submitting

Copy your project's URL and submit it below.