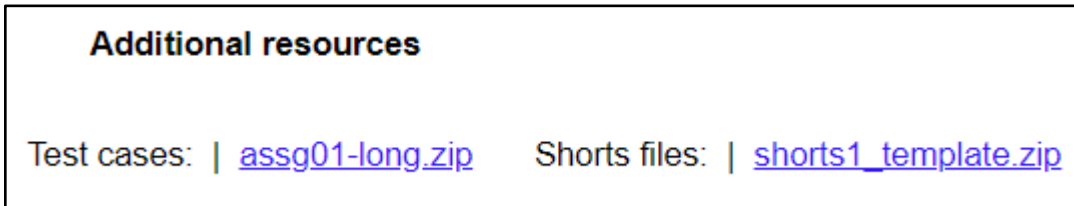


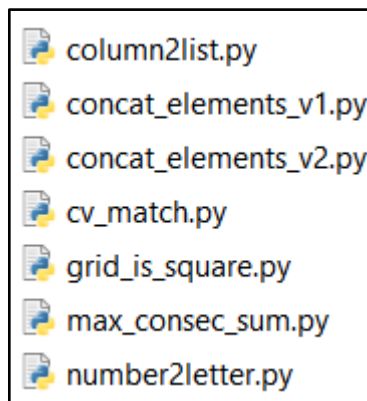
Submitting Short Problems to Gradescope

Step 1: Downloading the Shorts from the class website

1. Under the “Assignments” tab from your class website, look under the “Additional resources” column for the short problem template zip file.



2. Download this zip file and extract the contents. Inside the folder, you should see various python files—specifically one file for each short problem, with the same name as the short problem.

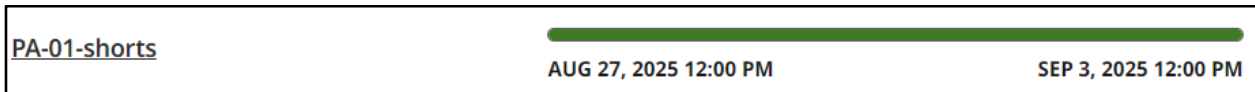


3. Open up VS Code or another IDE, and then add the folder with the short templates to your workspace. In VS Code, you would hit “File” > “Add Folder to Workspace” and then locate the folder you have downloaded.
4. Then, following the short problem instructions on the class website, you would fill out the rest of the starter code.

```
def column2list(grid, n):  
    # Your code goes here  
    pass
```

Step 2: Testing and Submitting on Gradescope

1. Locate the assignment in Gradescope labeled “PA-01-Shorts”, and click the link.



2. Submit the python files you have modified to run the autograder. If you are just trying to test specific short problems, you do not have to submit all of the short problem files for every submission—just the ones that you want to test. Make sure that your final submission **does** include all of the files though.

The screenshot shows the "Submit Programming Assignment" dialog box. It has a header "Submit Programming Assignment" and a sub-header "Upload all files for your submission". Under "Submission Method", there are three radio buttons: "Upload" (selected), "GitHub", and "Bitbucket". Below this, it says "Add files via Drag & Drop or [Browse Files](#)". There is a table with columns "Name", "Size", "Progress", and a close icon "x". The table lists several files with their sizes and progress bars, all of which are currently empty and marked with a red "x". At the bottom right, there are "Cancel" and "Upload" buttons.

Name	Size	Progress	x
column2list.py	60 b		x
concat_elements_v1.py	84 b		x
concat_elements_v2.py	85 b		x
cv_match.py	67 b		x
grid_is_square.py	63 b		x
max_consec_sum.py	66 b		x
number2letter.py	56 b		x

3. Once submitted, the autograder will run and automatically display which test cases you are passing or failing. If you are failing a test case, a message will be displayed showing you the inputs, expected outputs, and your output. If you are using something restricted for the short problem, the test case will fail and you will be told which content is banned. If your code results in a runtime error, the error message should be displayed.



If you have any comments, concerns, or general feedback about the autograder, feel free to let us know!