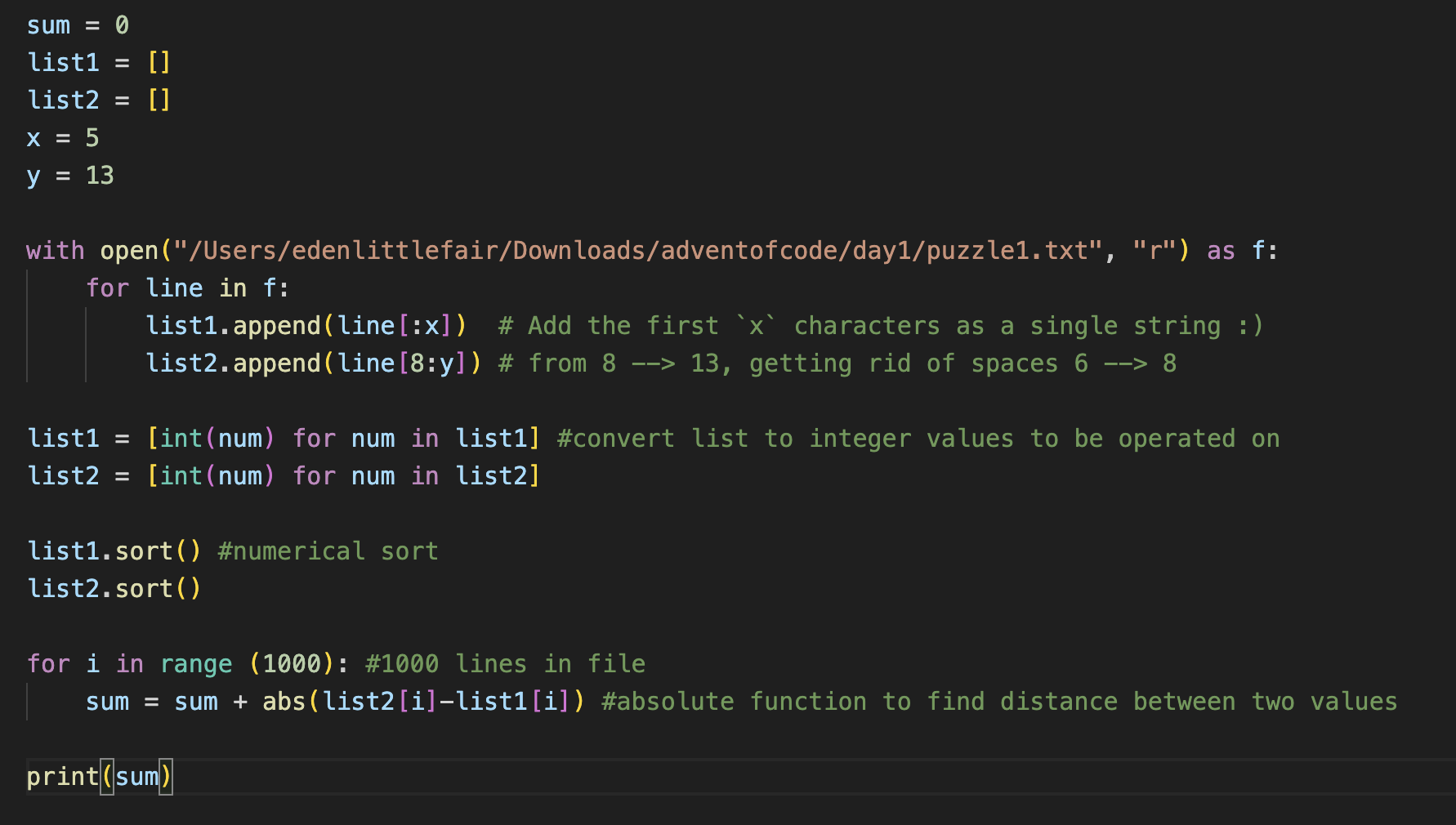
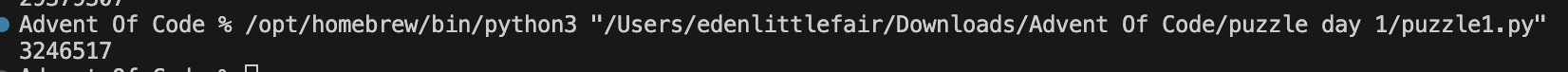
**Advent Of Code Log**

Challenge 1 part 1

Looked up how to add a range of characters from a file to a list, took a minute to get the right directory to the file. For the first list I looked at the first 5 characters in each line of the file, and for the second list I looked at characters 9-13 of the list, to avoid the 3 spaces in the middle. Then I looked up converting lists of string items into integers so that I could perform operations on them. Next I used the sort function (bit of a cheat) to sort the two lists in ascending order. Then I iterated through each item of the two lists simultaneously and found the absolute value between each of the two items in position i in each list, and summed them.

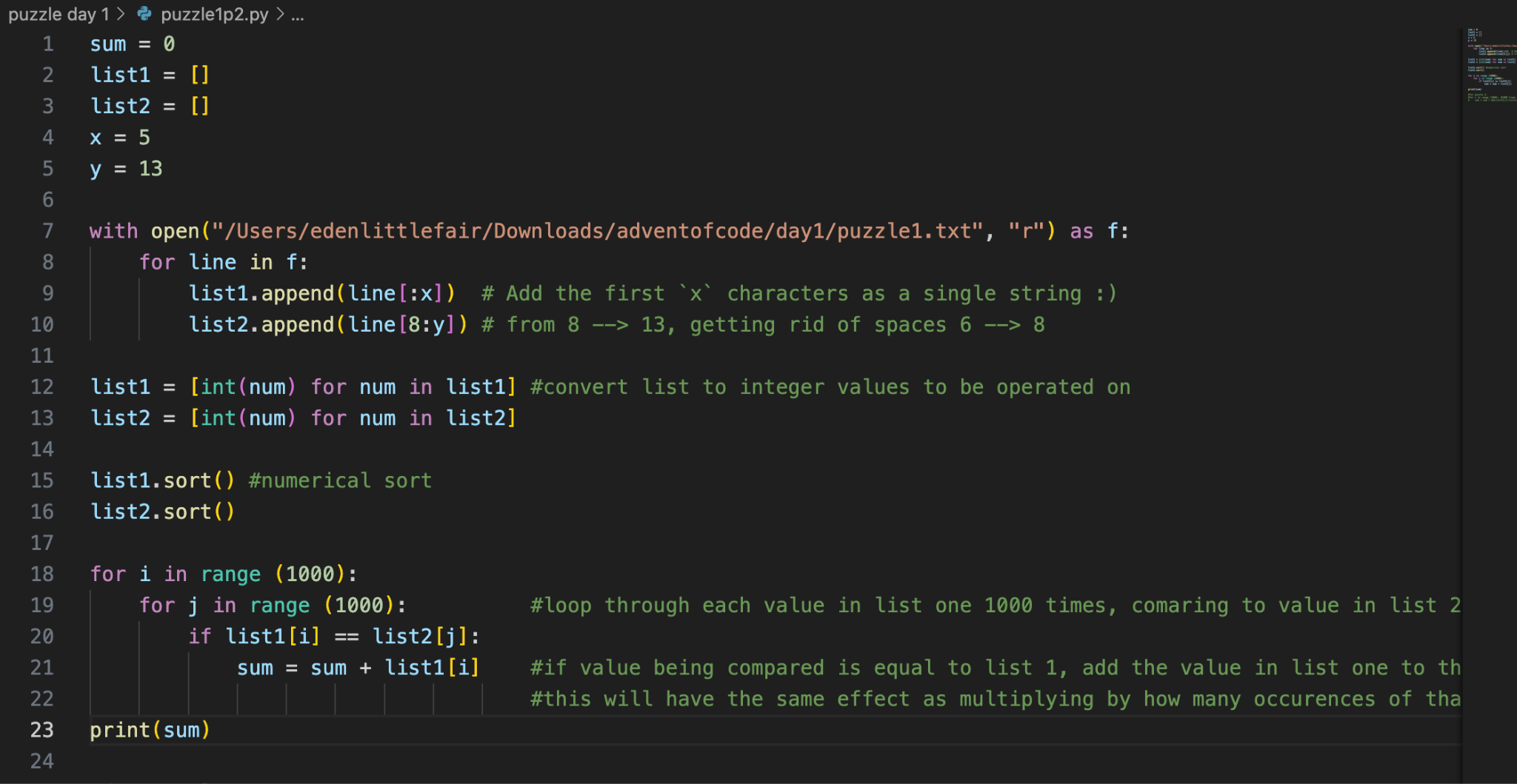


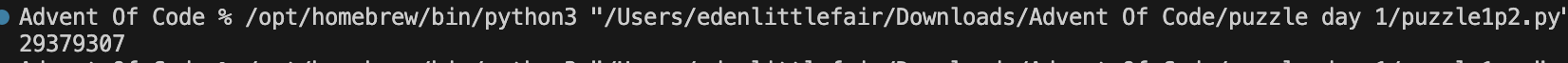


First answer i got out was correct

Challenge 1 part 2

Managed to solve this without looking anything up which I'm happy with, even if it wasn't that difficult, hopefully showing me that I'm improving at the syntactual side of python. Basically only changed lines 18-21 of the program, rather than finding the difference between the two items in position i, you now have to select each item one by one in list 1 and compare it to every single item in list 2, which I did using a nested for loop. In terms of the logic, the question is asking to find how many times the item (i) in list one appears in list two, and then do the value of that item (i) multiplied by the number of times it appears in list 2. To avoid this multiplication step, I added the item in list 1 to the sum every time it appeared in list 2. That then iterates through a total of 1 million times, not the best time complexity, but it works.





First answer i got out was correct

Challenge 2 part 1

I am actually super happy I managed to get this, and using my own logic as well, miraculously on the first try. I kinda got lost a bit with this one with so many selection statements I had to keep backtracking to see what I was actually doing. One thing I got stuck on for a minute was only incrementing the total by one if the entire line was true, and not just adjacent pairs. I ended up getting a load of outputs that were greater than 1000 because I was adding one each time the conditions were met by adjacent pairs, rather than the whole line. I fixed this by having a counter as well as a total, so that if the counter equalled the length of the row minus 1 (i.e true for all adjacent pairs of numbers) then it would add one to the sum, which was the overall counter for which rows were ‘safe’ and which weren’t. Definitely could have used better variable names for sum and total, total should probably have been counter and sum should have been total. I approached the problem using 2d arrays, once again looking up how to get the text file in the form of a 2d array for me to work on it. (line 4 → 7).There is a decent amount of logic here, mostly lines 11 → 18 and then lines 20 → 27 is just repeated for rows that are decreasing.

Conditions:

-Rows have to either always be increasing or always be decreasing

-The distance between two values cannot be greater than 3, and cannot be less than 1

Breakdown of lines 11-18:

Line 11: initialising first if statement for lists that are increasing, by comparing the first two items in the list/array (is it an array or a list?!)

Line 12: setting the counter to 0, used to check if every condition was met for each adjacent pair at the end

Line 13 → 14: Initialising a for loop to iterate through each adjacent pair and make sure the condition is met that the integer before is less than the next integer

Line 15: other condition for question

Line 16: if all conditions are met for that one pair, the total is then incremented by one

Line 17-18: I just realised this is unnecessarily checking if every adjacent pair meets the required conditions each loop, when you only need to check at the end, but it doesn't matter. This is checking right at the end if every single adjacent pair held true to the conditions, and if they did the sum will be incremented by one. If not, it will start the loop again and the total will reset to 0. If they all met the required conditions then the value of total will be len(list1[i]) - 1





First answer i got out was correct

Challenge 2 part 2

I got cooked, but still cooked.

will explain more tomorrow

