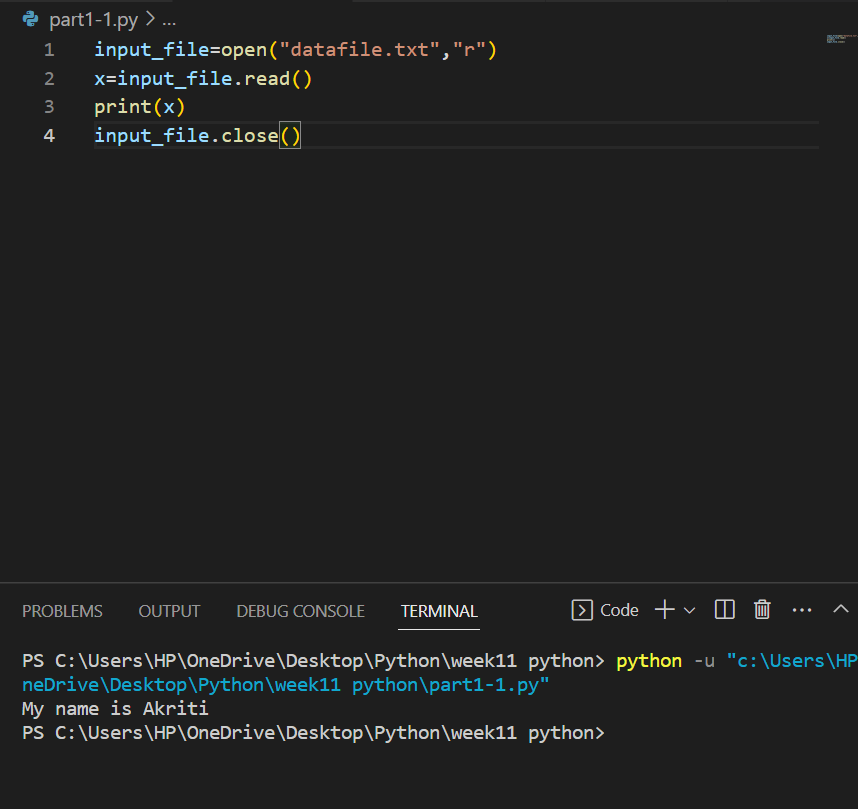
**4CS001**

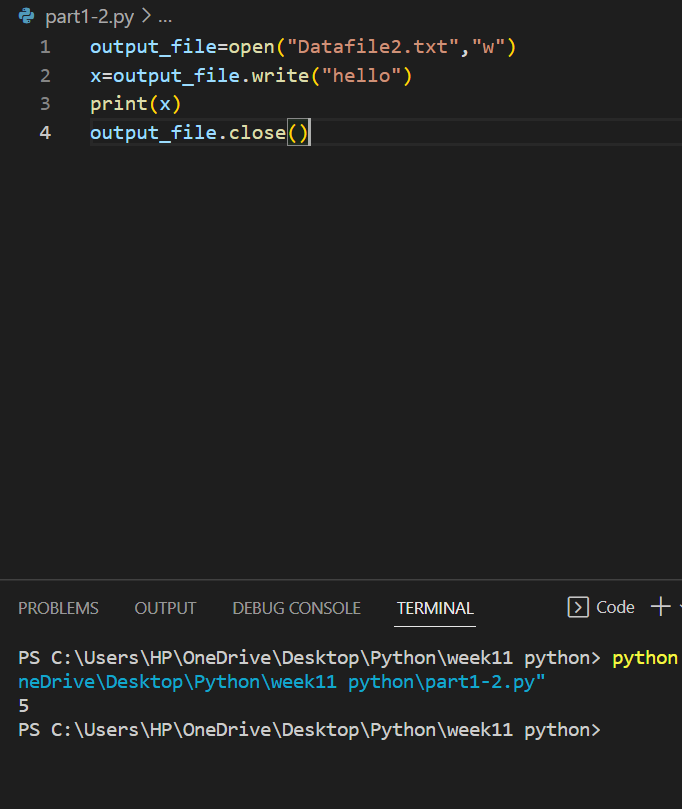
**Python Workshop 11: File Handling**

**Part 1**

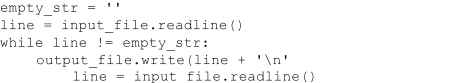
1. Create a program in Python that opens a file named 'datafile.txt' for reading and assigns identifier input\_file to the file object created.

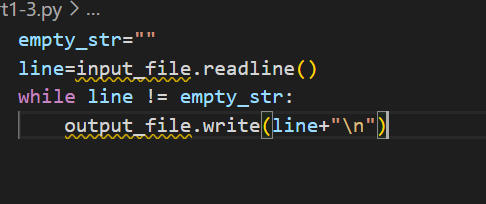


1. Create a program in Python that opens a file named 'datafile2.txt' for writing and assigns identifier output\_file to the file object created.



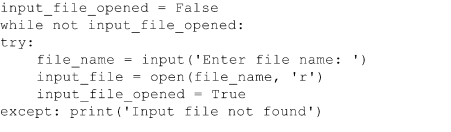
3. Assume that input\_file is a file object for a text file open for reading, and output\_file is a file object for a text file open for writing. Explain the contents of the output after the following code terminates:



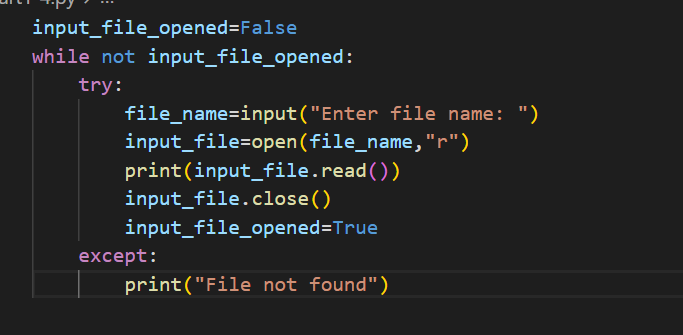


=Here, The loop runs as long as the line is not equal to empty string. Here, readline() reads the file line by line and this gets written to another file with object output\_file as long as the condition is met.

4. Identify the error in the following code:



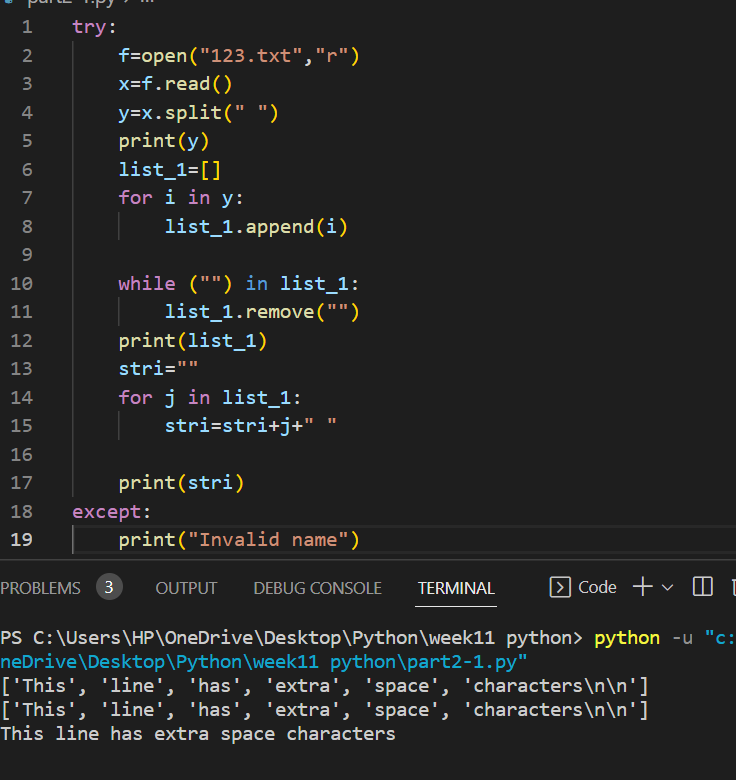
=The given code does not close the opened file so here is the code without errors.



**Part 2**

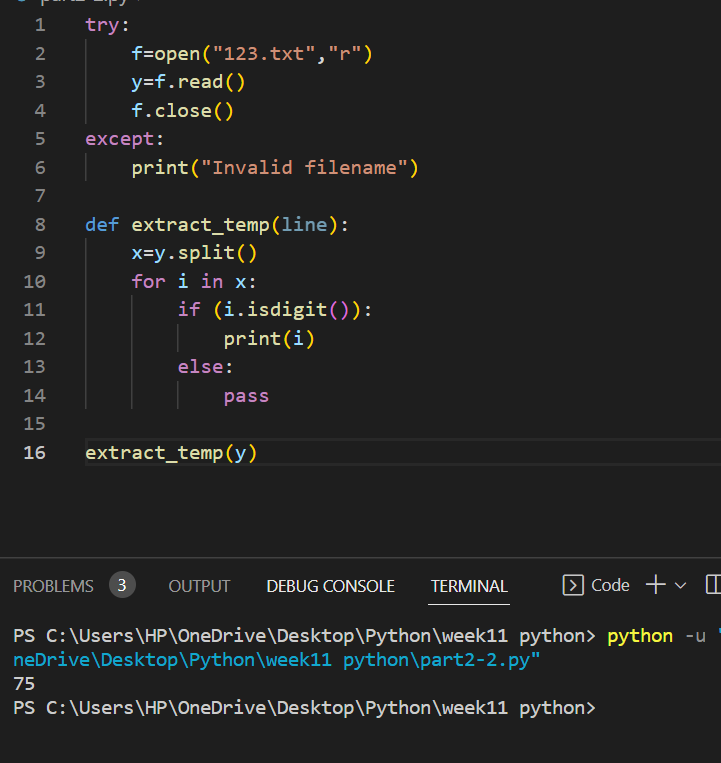
1. Write a Python function called reduce\_spaces that is given a line read from a text file and returns the line with all extra space characters removed:

‘This line has extra space characters’ → ’This line has extra space characters’



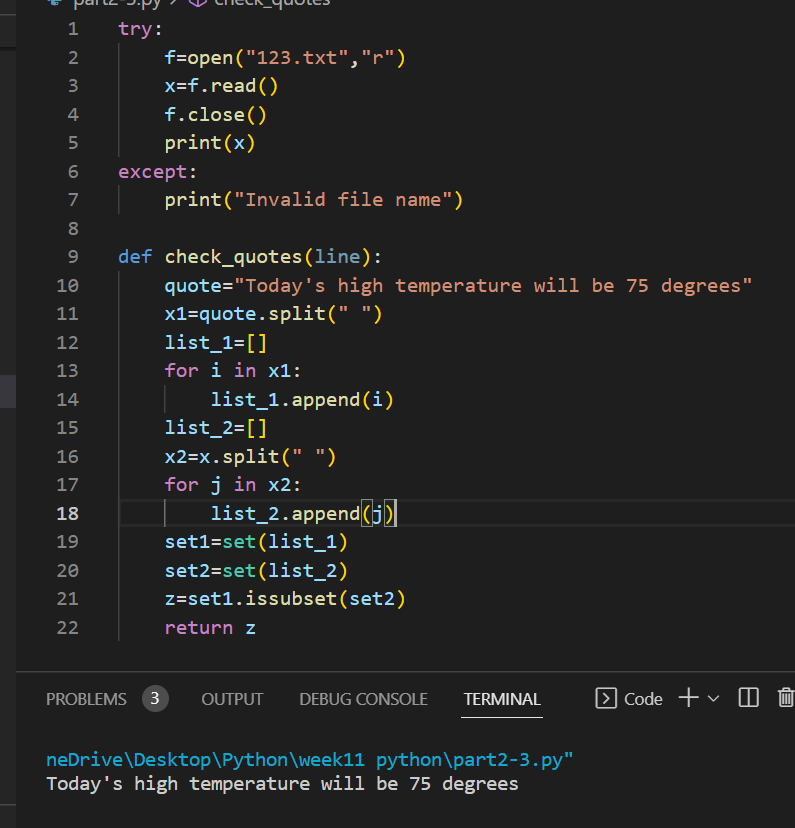
2. Write a Python function named extract\_temp that is given a line read from a text file and displays the one number (integer) found in the string:

’The high today will be 75 degrees’ → 75.



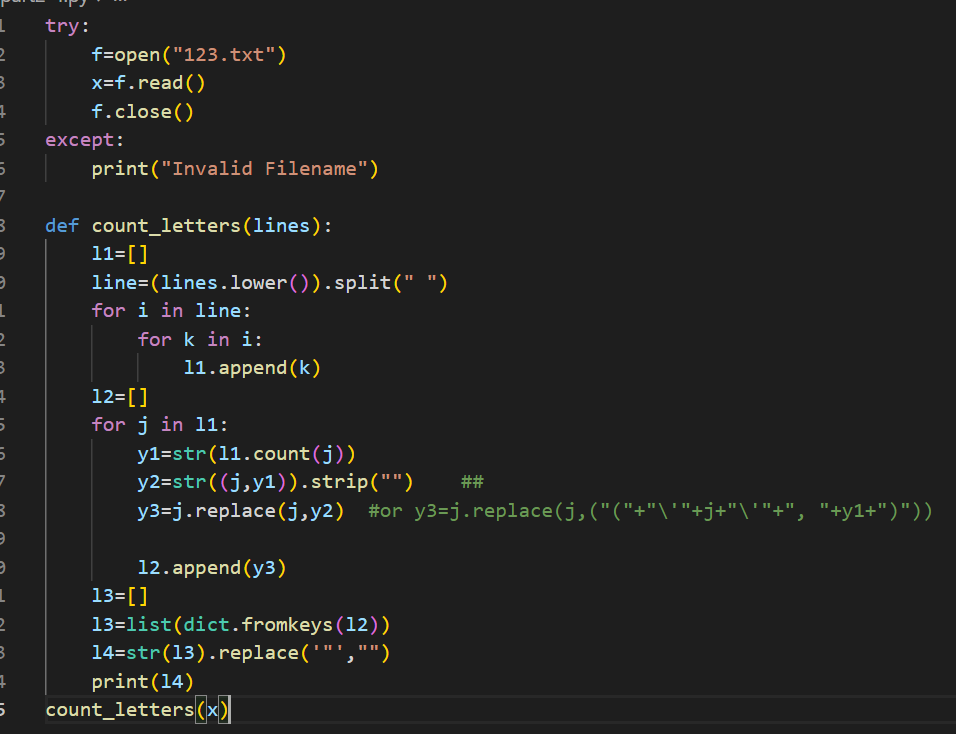
3. Write a Python function named check\_quotes that is given a line read from a text file and returns True if each quote characters in the line has a matching quote (of the same type), otherwise returns False.

‘Today’s high temperature will be 75 degrees’ → False



4. Write a Python function named count\_letters that is given a line read from a text file and returns a list containing every letter in the line and the number of times that each letter appears (with upper/lower case letters counted together)

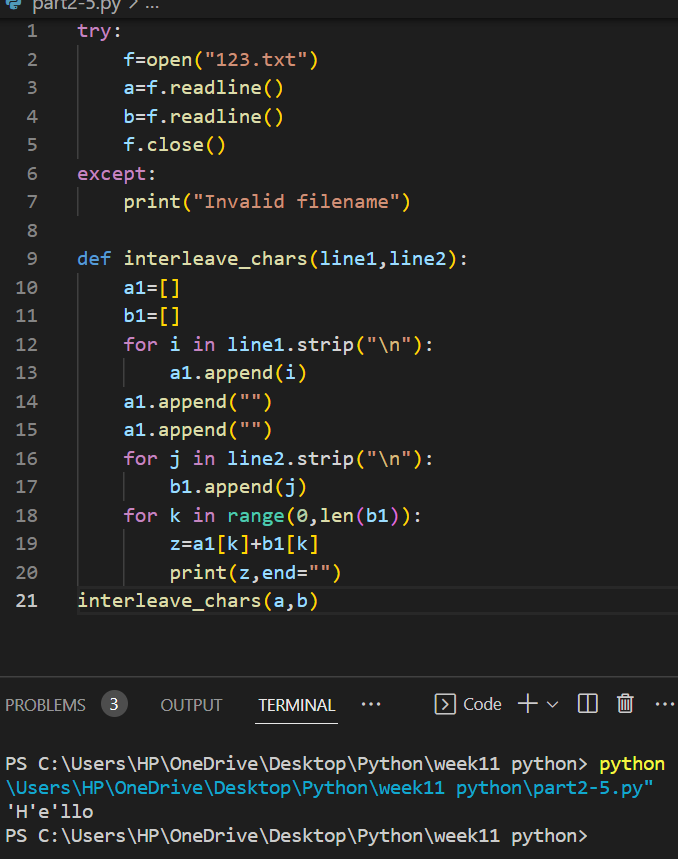
‘This is a line’ → [ (‘t’, 1), (‘h’, 1), (‘i’, 3), (‘s’, 2), (‘a’, 1), (‘l’, 1), (‘n’, 1), (‘e’, 1) ]



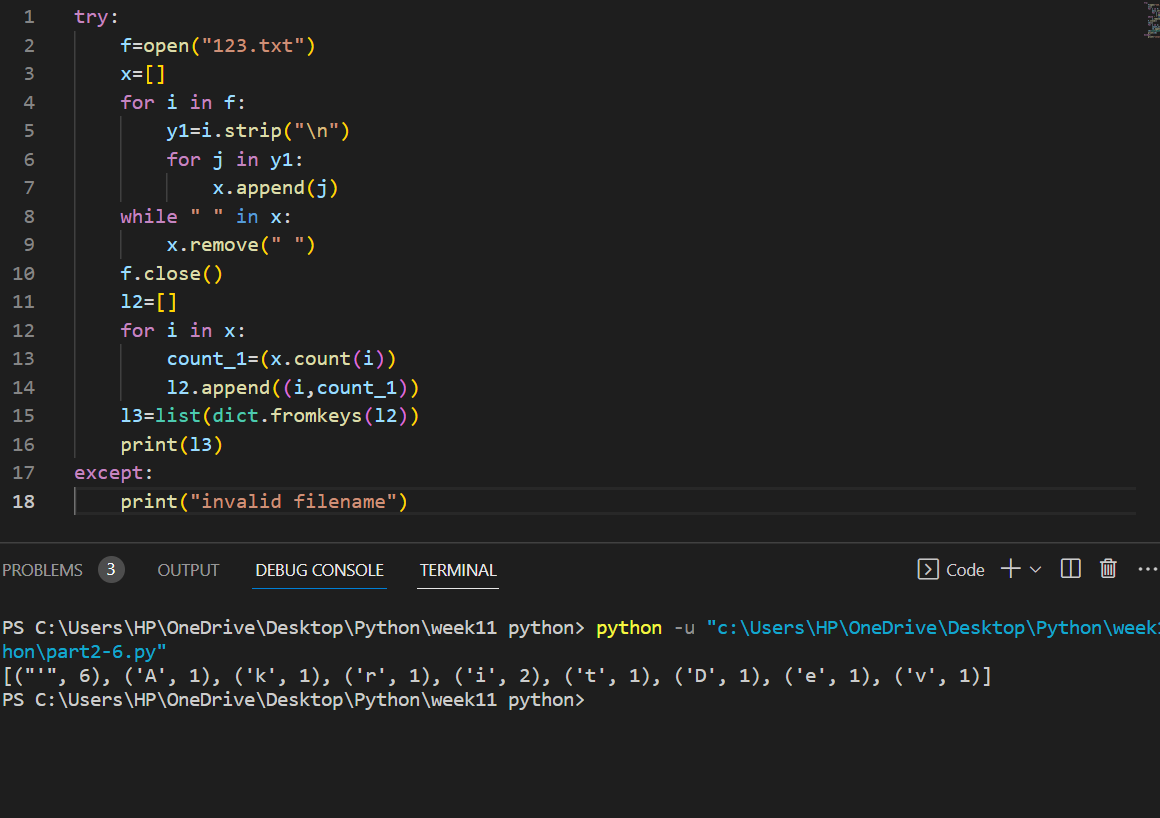
Text

Description automatically generated

5. Write a Python function named interleave\_chars that is given two lines read from a text, and returns a single string containing the characters of each string interleaved: ‘Hello’, ‘Goodbye’ → ‘HGeololdobye’



6. Give a for loop that counts all the characters in a string assigned to variable line, except blanks and the newline character.



7. For variable month which contains the full name of any given month, give an expression to display just the first three letters of the month.

Text

Description automatically generated

8. Give an expression that displays True if the letter ‘r’ appears in a given month name stored in variable month, otherwise displays False.

Graphical user interface, text, application

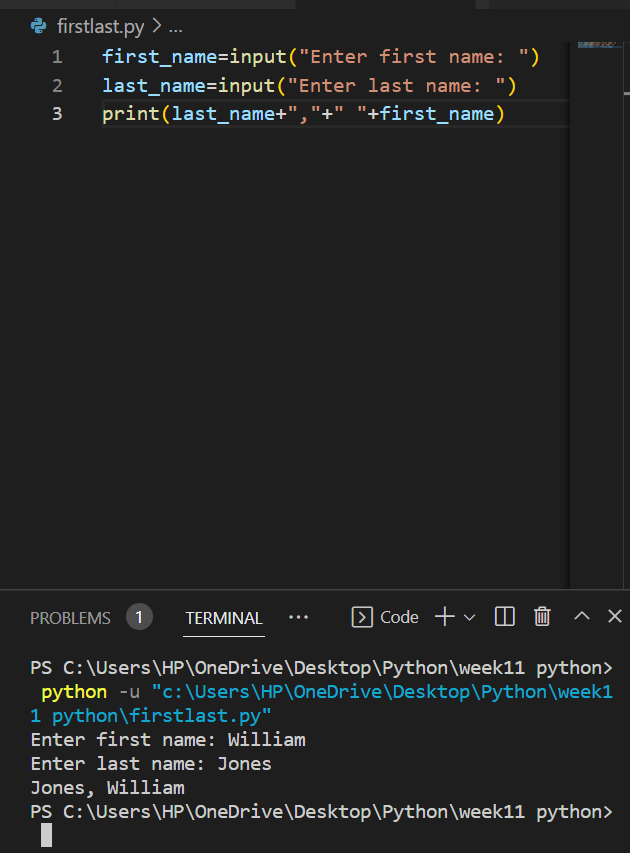
Description automatically generated

9. Give an expression for determining how many times the letter ‘r’ appears in a given month name stored in variable month.

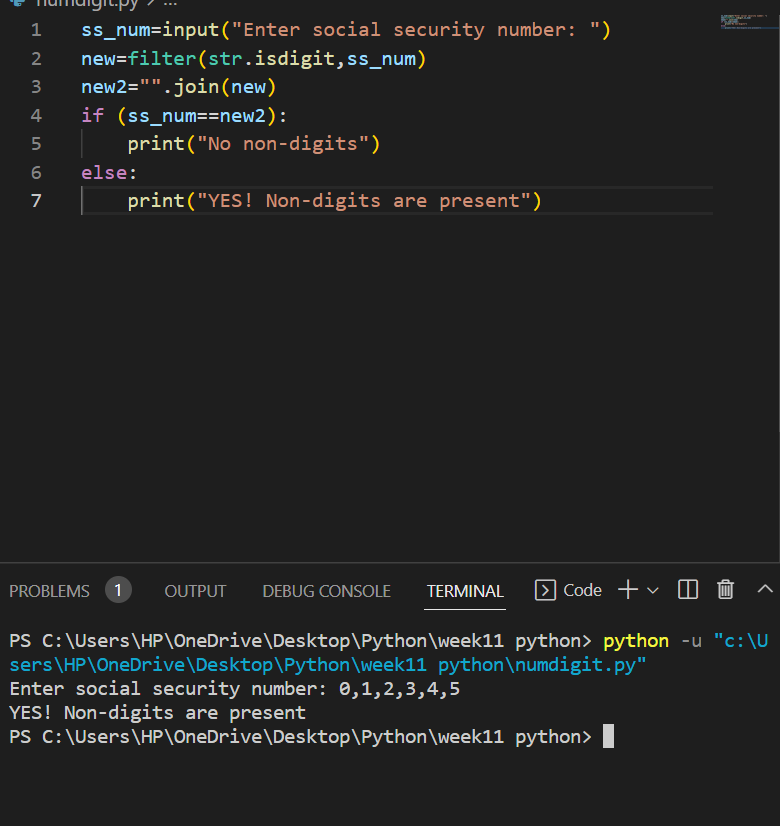
Graphical user interface, text

Description automatically generated

10. For a person’s first name stored in variable first\_name, and last name stored in variable last\_name, give an expression that displays the person’s name formatted exactly as follows: Jones, William.



11. Give an instruction that determines if a given social security number represented as a string and stored in variable ss\_num, contains any non- digits.



12. Give an instruction that determines the index of the ‘@’ character in an email address stored in variable email\_addr.

Text

Description automatically generated

13. For a variable named date containing a date in the form 12/14/2012, give an expression that replaces all slashes characters with dashes.

Text

Description automatically generated

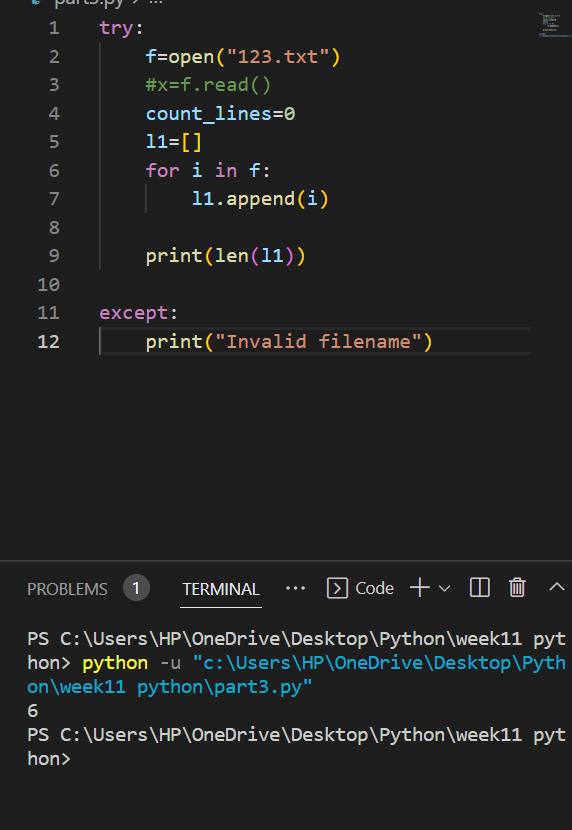
14. For a variable named err\_mesg that contains error messages in the form \*\* error message \*\*, give an expression that produces a string containing the error message without the leading and trailing asterisks and blank characters.

Text

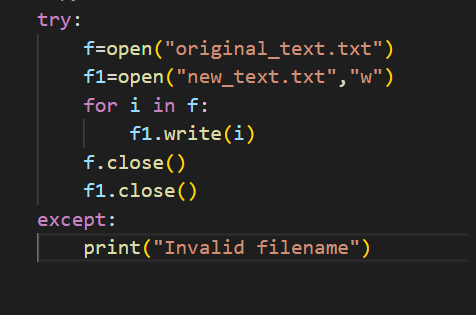
Description automatically generated

**Part 3**

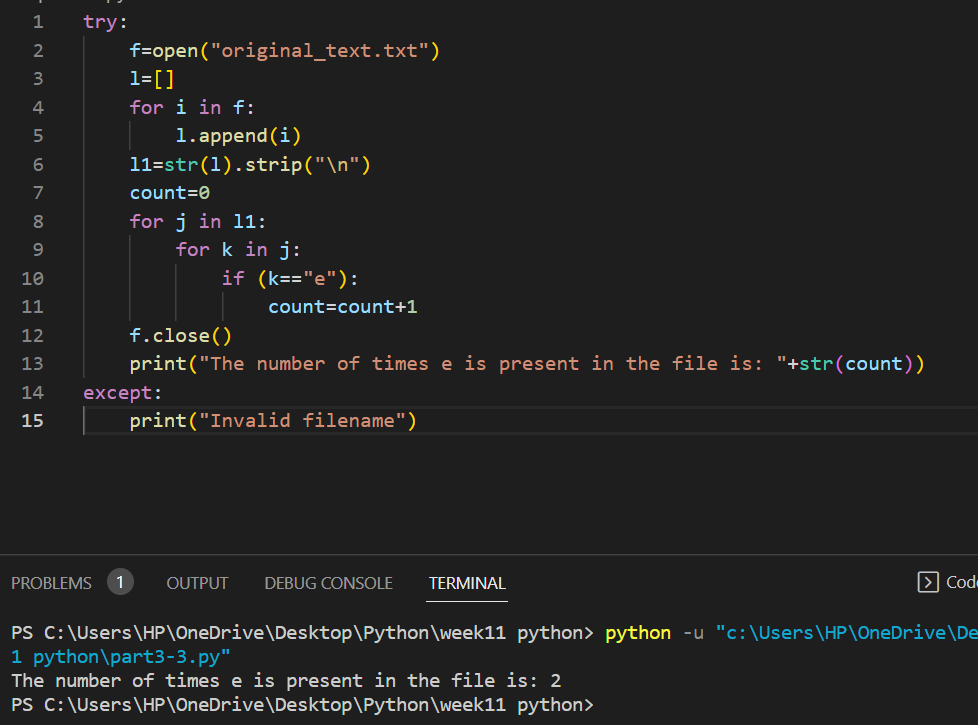
1. Write a program that opens and reads a text file and displays how many lines of text are in the file.



1. Write a program that reads a text file named original\_text, and writes every other line, starting with the first line, to a new file named new\_text.

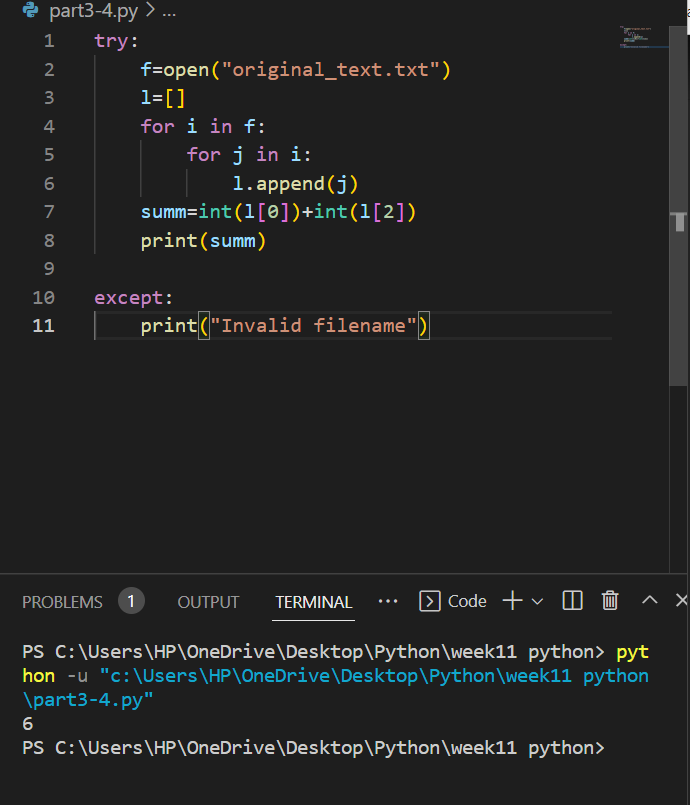


1. Write a program that reads a text file named original\_text, and counts how many time the letter 'e' occurs (the most frequently occurring letter in English), and displays how many occurrences there are.



4. Write a program that reads a text file containing numerical expressions on

each line and print them out along with the results. For example, for the numerical expression 4 + 2 in your file, your program should output: 4 + 2 = 6.

****

**Part 4 (Optional)**

Write a Python program that encrypts and decrypts text files using a substitution cipher. Your program should ask the user for the name of a text file and whether they would like to encrypt or decrypt. Once the process is complete, you should write the output to a new text file with a modified name:

This program will encrypt and decrypt text files

Enter (e) to encrypt a password, and (d) to decrypt: e

Enter the name of a text file to encrypt: hello.txt Output written to: encrypted\_hello.txt

Your program should catch exceptions and print helpful error messages. You should use your solution to Coding Challenge 03 to help you.