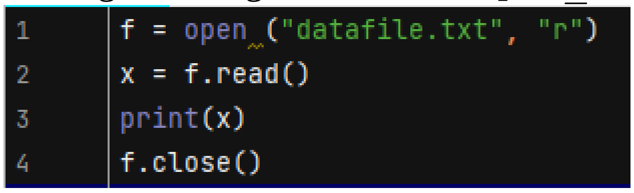
4CS001

Python Workshop 11: File Handling

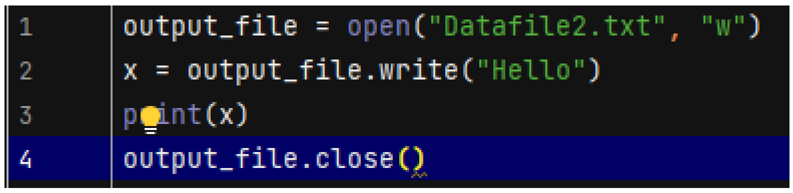
Part 1

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.



```
f = open ("datafile.txt", "r")
x = f.read()
print(x)
f.close()
```

2. Create a program in Python that opens a file named `'datafile2.txt'` for writing and assigns identifier `output_file` to the file object created.



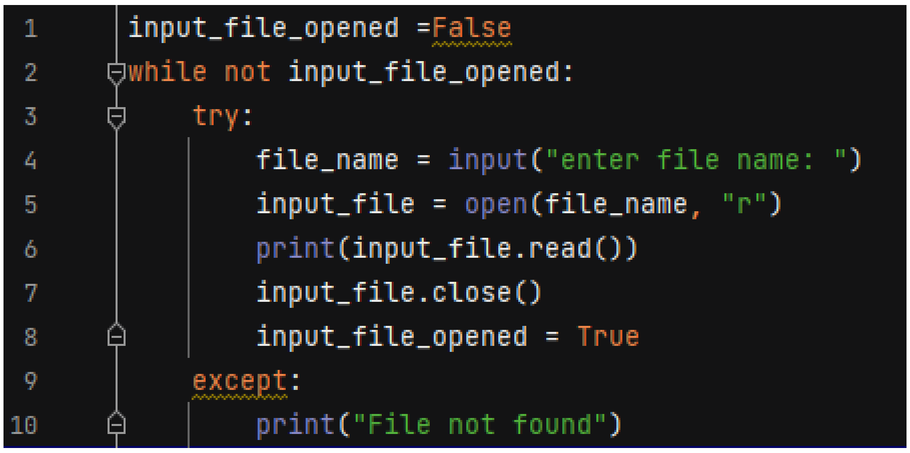
```
output_file = open("Datafile2.txt", "w")
x = output_file.write("Hello")
print(x)
output_file.close()
```

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:

```
empty_str = ''
line = input_file.readline()
while line != empty_str:
    output_file.write(line + '\n'
        line = input file.readline()
```

4. Identify the error in the following code:

```
input_file_opened = False
while not input_file_opened:
try:
    file_name = input('Enter file name: ')
    input_file = open(file_name, 'r')
    input_file_opened = True
except: print('Input file not found')
```

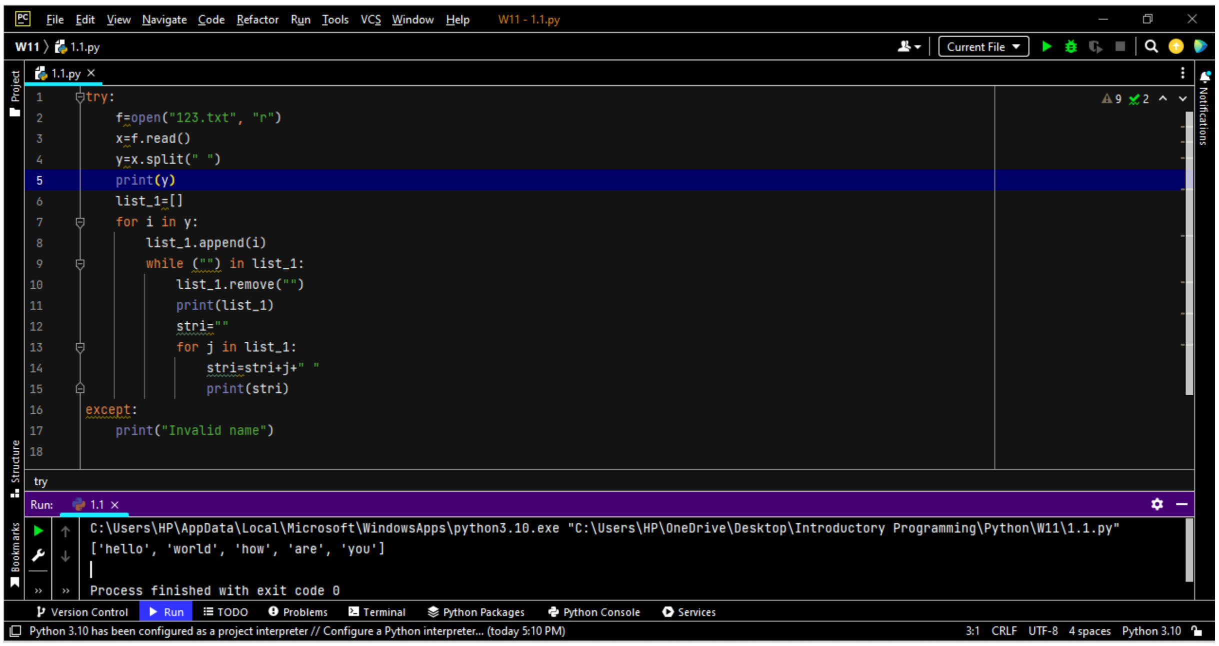


```
input_file_opened =False
while not input_file_opened:
    try:
        file_name = input("enter file name: ")
        input_file = open(file_name, "r")
        print(input_file.read())
        input_file.close()
        input_file_opened = True
    except:
        print("File not found")
```

<u>Part 2</u>

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’



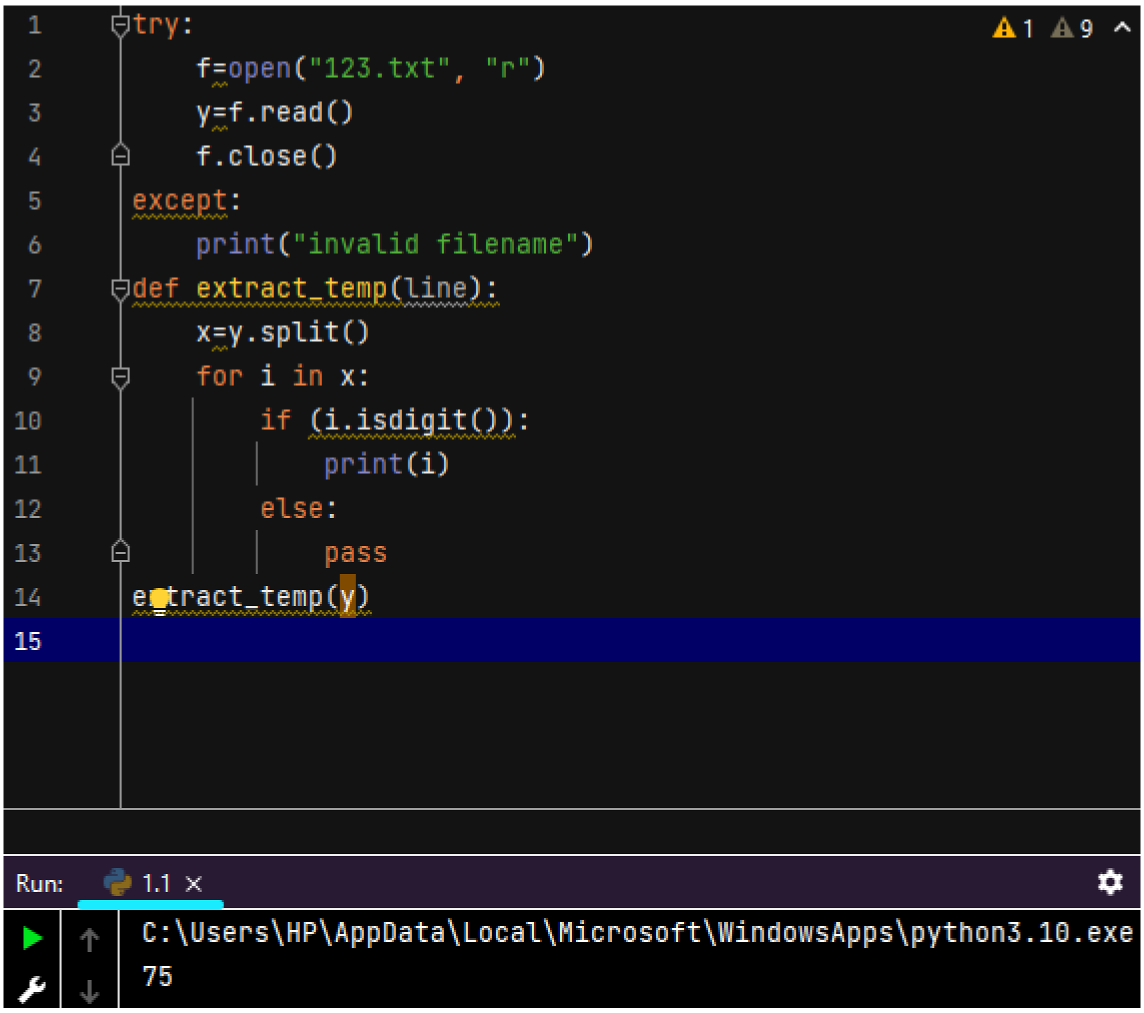
File Edit View Navigate Code Refactor Run Tools VCS Window Help
W11 - 1.1.py
W11 〉 1.1.py
Current File
1.1.py
try
Run: 1.1
Version Control Run TODO Problems Terminal Python Packages Python Console Services
Python 3.10 has been configured as a project interpreter // Configure a Python interpreter... (today 5:10 PM)
3:1 CRLF UTF-8 4 spaces Python 3.10

```
try:
    f=open("123.txt", "r")
    x=f.read()
    y=x.split(" ")
    print(y)
    list_1=[]
    for i in y:
        list_1.append(i)
        while ("") in list_1:
            list_1.remove("")
            print(list_1)
            stri=""
            for j in list_1:
                stri=stri+j+" "
                print(stri)
except:
    print("Invalid name")
```

```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\python3.10.exe "C:\Users\HP\OneDrive\Desktop\Introductory Programming\Python\W11\1.1.py"
['hello', 'world', 'how', 'are', 'you']
|
Process finished with exit code 0
```

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.

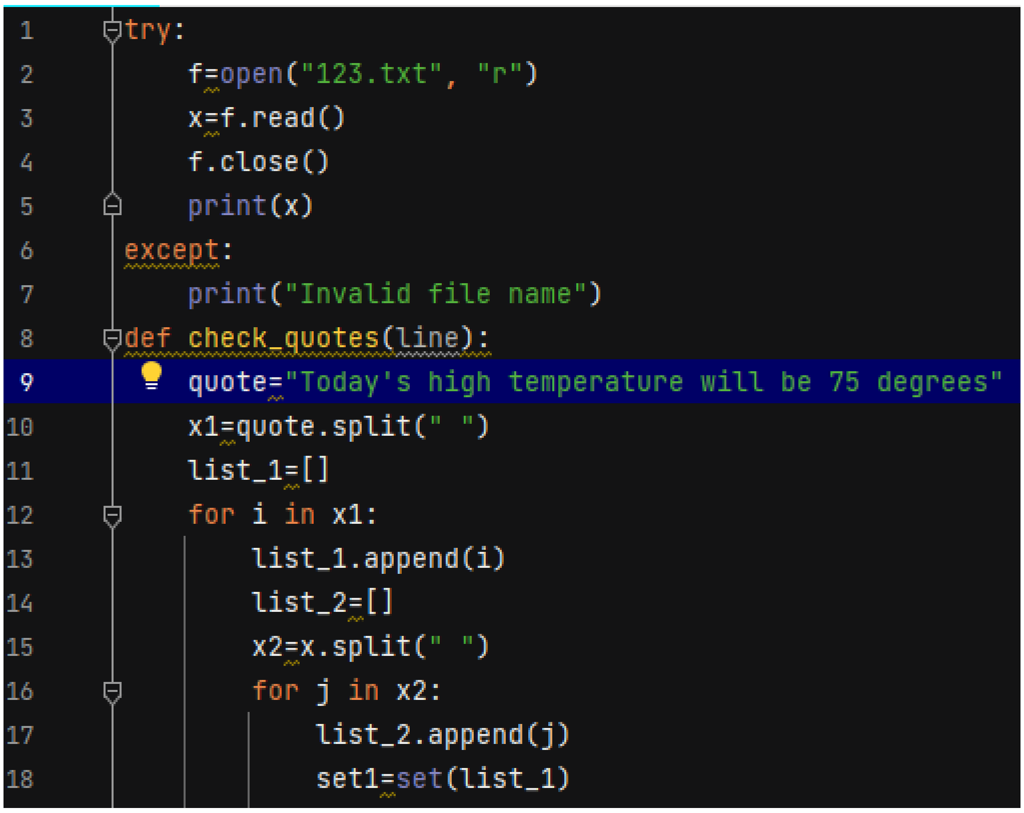


```
try:
    f=open("123.txt", "r")
    y=f.read()
    f.close()
except:
    print("invalid filename")
def extract_temp(line):
    x=y.split()
    for i in x:
        if (i.isdigit()):
            print(i)
        else:
            pass
extract_temp(y)
```

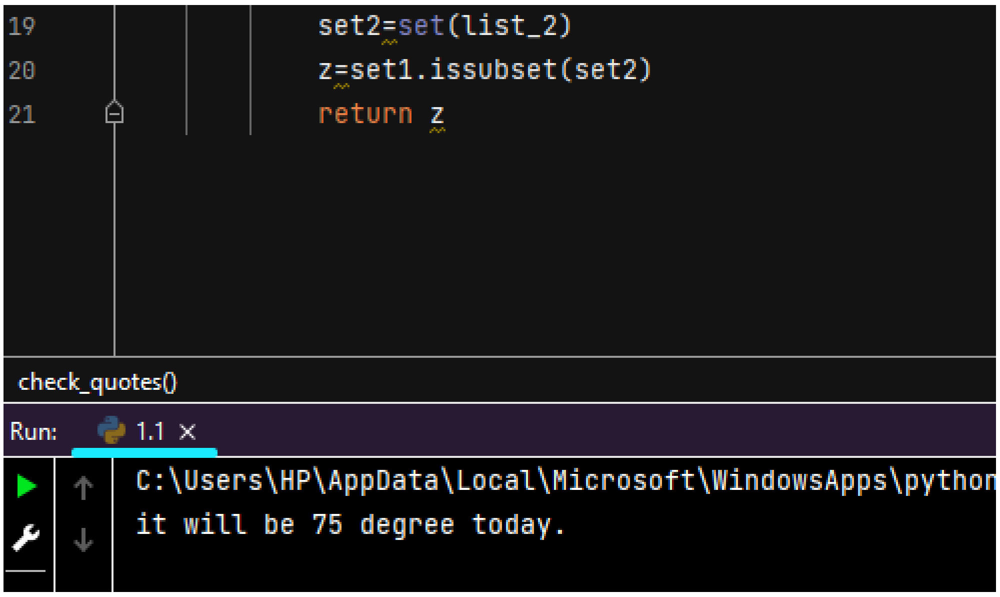
```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\python3.10.exe
75
```

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



```python
try:
    f=open("123.txt", "r")
    x=f.read()
    f.close()
    print(x)
except:
    print("Invalid file name")
def check_quotes(line):
    quote="Today's high temperature will be 75 degrees"
    x1=quote.split(" ")
    list_1=[]
    for i in x1:
        list_1.append(i)
        list_2=[]
        x2=x.split(" ")
        for j in x2:
            list_2.append(j)
            set1=set(list_1)
```



```python
            set2=set(list_2)
            z=set1.issubset(set2)
            return z
```

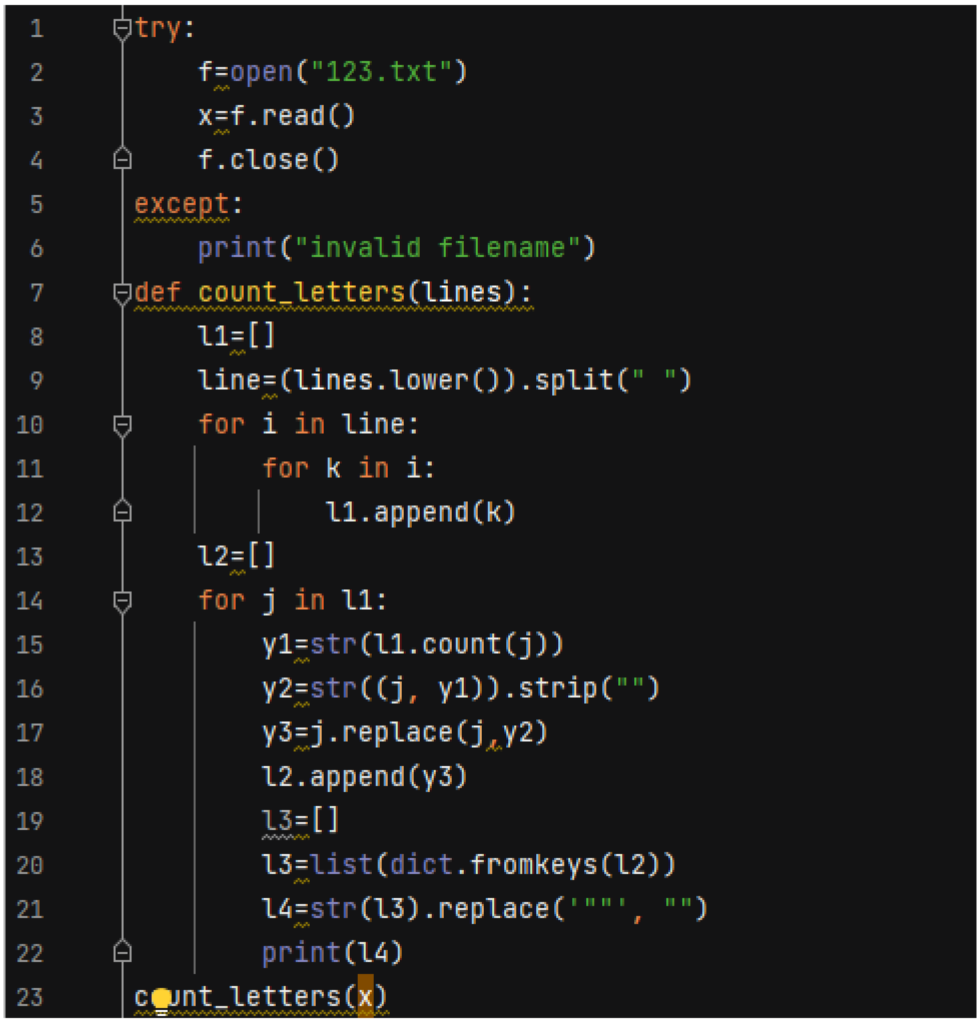
check_quotes()

Run: 1.1

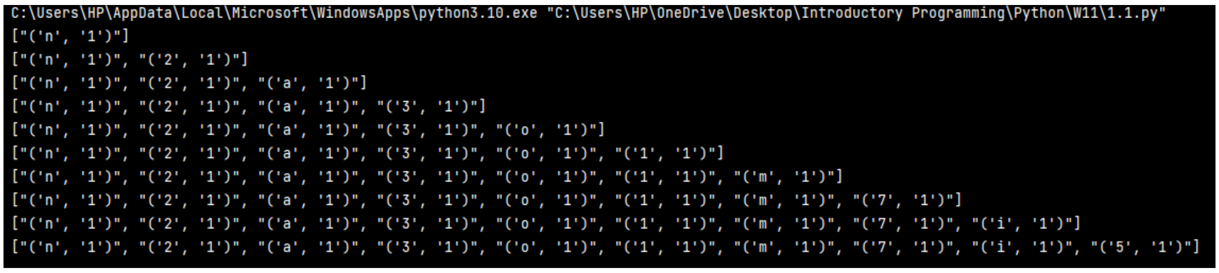
```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\python
it will be 75 degree today.
```

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)]

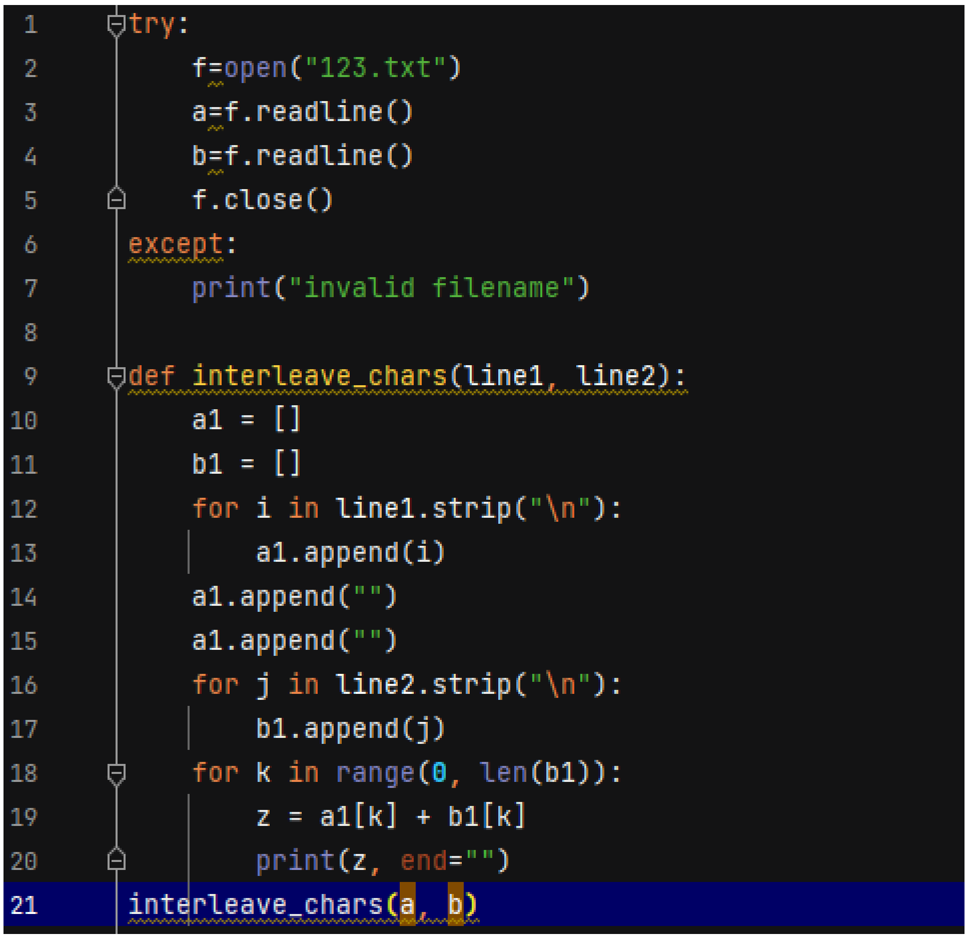


```python
try:
    f=open("123.txt")
    x=f.read()
    f.close()
except:
    print("invalid filename")
def count_letters(lines):
    l1=[]
    line=(lines.lower()).split(" ")
    for i in line:
        for k in i:
            l1.append(k)
    l2=[]
    for j in l1:
        y1=str(l1.count(j))
        y2=str((j, y1)).strip("")
        y3=j.replace(j,y2)
        l2.append(y3)
        l3=[]
        l3=list(dict.fromkeys(l2))
        l4=str(l3).replace('""', "")
        print(l4)
count_letters(x)
```

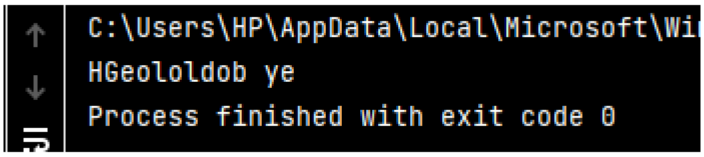


```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\python3.10.exe "C:\Users\HP\OneDrive\Desktop\Introductory Programming\Python\W11\1.1.py"
["('n', '1')"]
["('n', '1')", "('2', '1')"]
["('n', '1')", "('2', '1')", "('a', '1')"]
["('n', '1')", "('2', '1')", "('a', '1')", "('3', '1')"]
["('n', '1')", "('2', '1')", "('a', '1')", "('3', '1')", "('o', '1')"]
["('n', '1')", "('2', '1')", "('a', '1')", "('3', '1')", "('o', '1')", "('1', '1')"]
["('n', '1')", "('2', '1')", "('a', '1')", "('3', '1')", "('o', '1')", "('1', '1')", "('m', '1')"]
["('n', '1')", "('2', '1')", "('a', '1')", "('3', '1')", "('o', '1')", "('1', '1')", "('m', '1')", "('7', '1')"]
["('n', '1')", "('2', '1')", "('a', '1')", "('3', '1')", "('o', '1')", "('1', '1')", "('m', '1')", "('7', '1')", "('i', '1')"]
["('n', '1')", "('2', '1')", "('a', '1')", "('3', '1')", "('o', '1')", "('1', '1')", "('m', '1')", "('7', '1')", "('i', '1')", "('5', '1')"]
```

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’



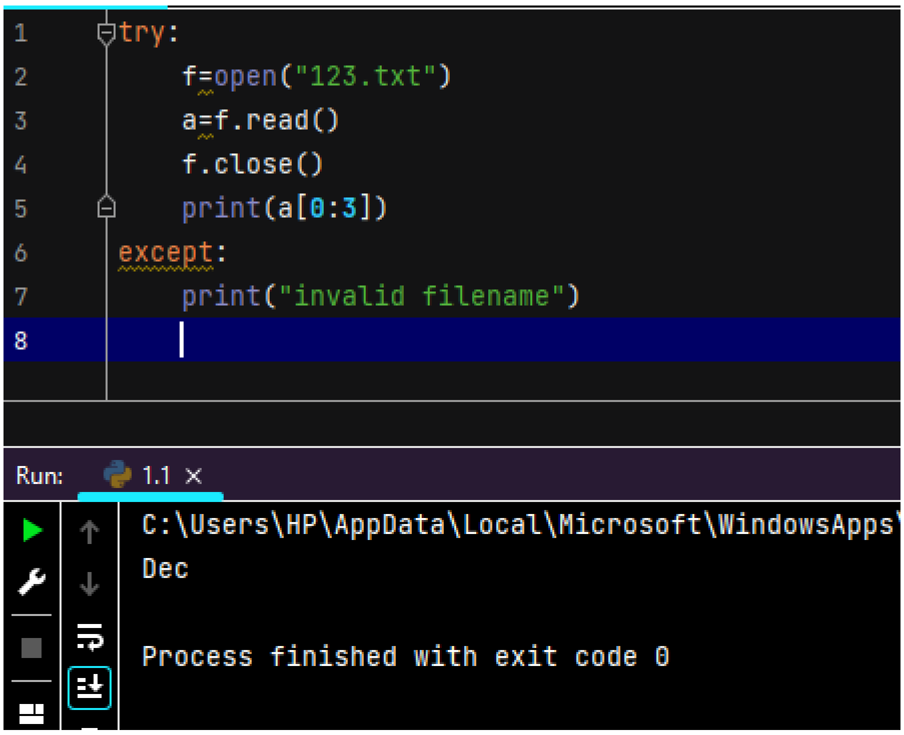
```python
try:
    f=open("123.txt")
    a=f.readline()
    b=f.readline()
    f.close()
except:
    print("invalid filename")

def interleave_chars(line1, line2):
    a1 = []
    b1 = []
    for i in line1.strip("\n"):
        a1.append(i)
    a1.append("")
    a1.append("")
    for j in line2.strip("\n"):
        b1.append(j)
    for k in range(0, len(b1)):
        z = a1[k] + b1[k]
        print(z, end="")
interleave_chars(a, b)
```



```
C:\Users\HP\AppData\Local\Microsoft\Wi
HGeololdob ye
Process finished with exit code 0
```

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.

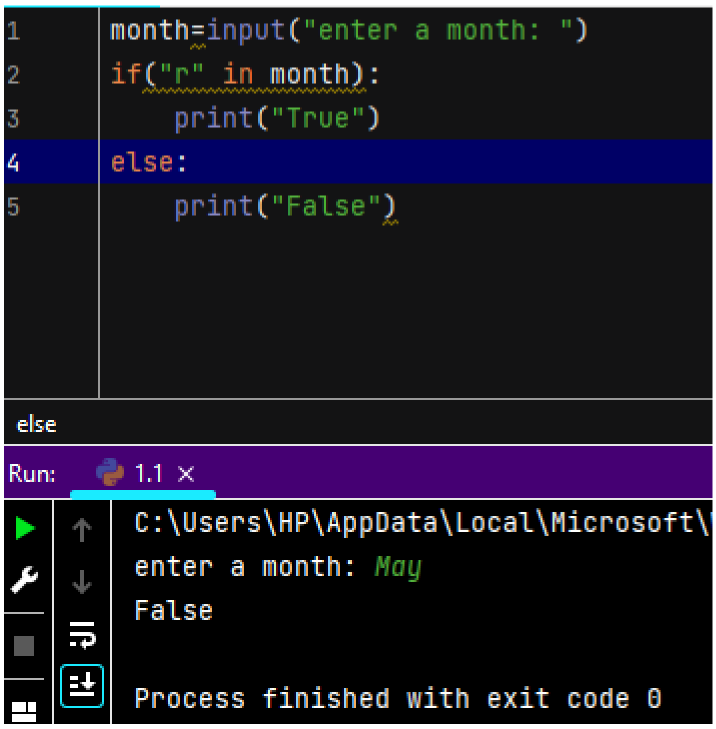


```
try:
    f=open("123.txt")
    a=f.read()
    f.close()
    print(a[0:3])
except:
    print("invalid filename")

```

Run: 1.1 ×

```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps
Dec

Process finished with exit code 0
```

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



```
month=input("enter a month: ")
if("r" in month):
    print("True")
else:
    print("False")
```

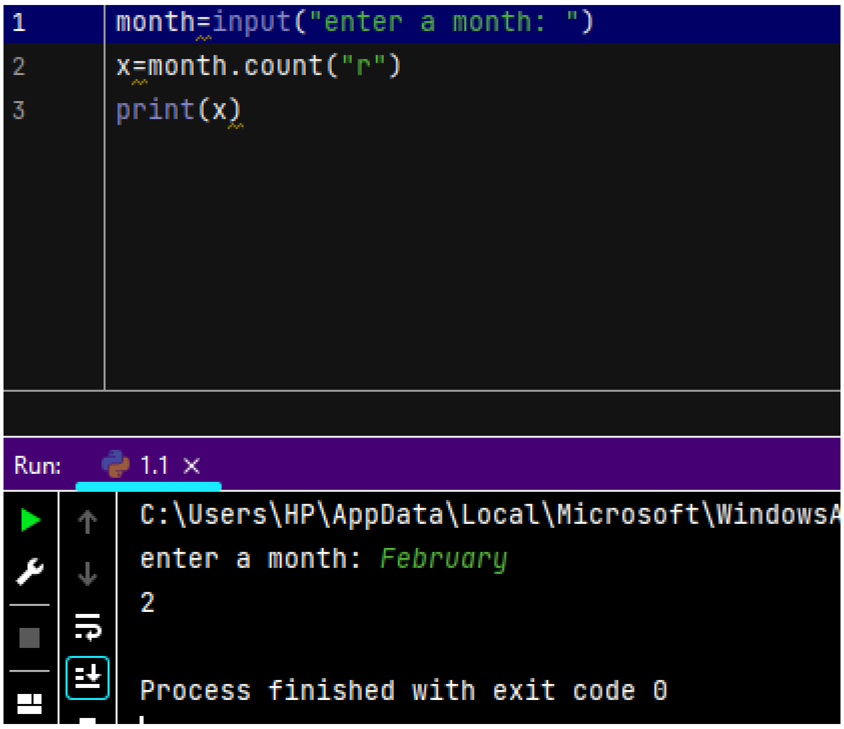
else

Run: 1.1 ×

```
C:\Users\HP\AppData\Local\Microsoft\
enter a month: May
False

Process finished with exit code 0
```

9. Give an expression for determining how many times the letter ‘r’ appears in a

given month name stored in variable `month`.

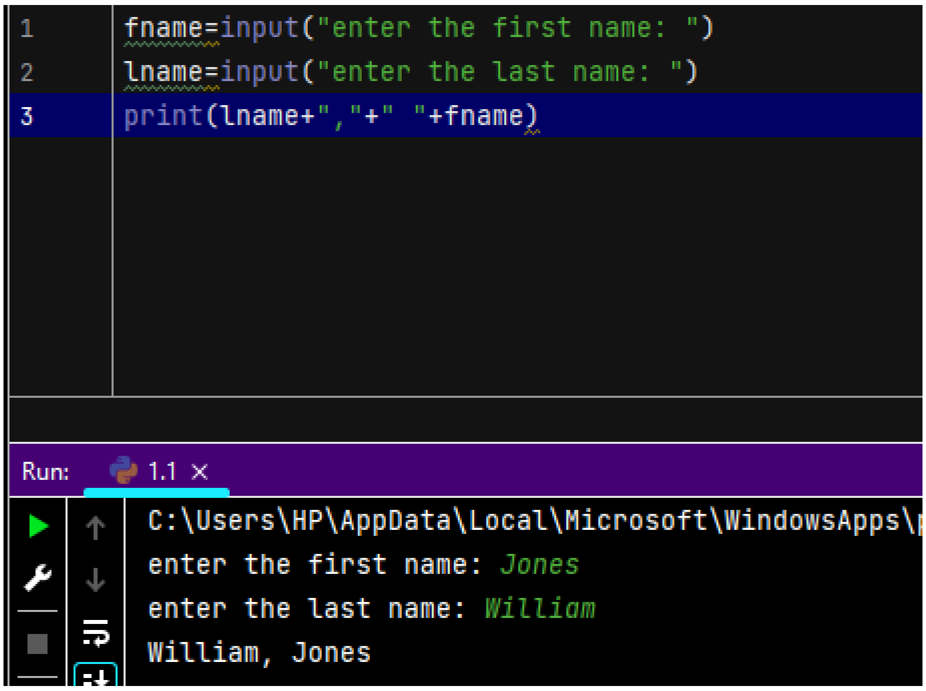


```
month=input("enter a month: ")
x=month.count("r")
print(x)
```

Run: 1.1

```
C:\Users\HP\AppData\Local\Microsoft\WindowsA
enter a month: February
2

Process finished with exit code 0
```

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`.

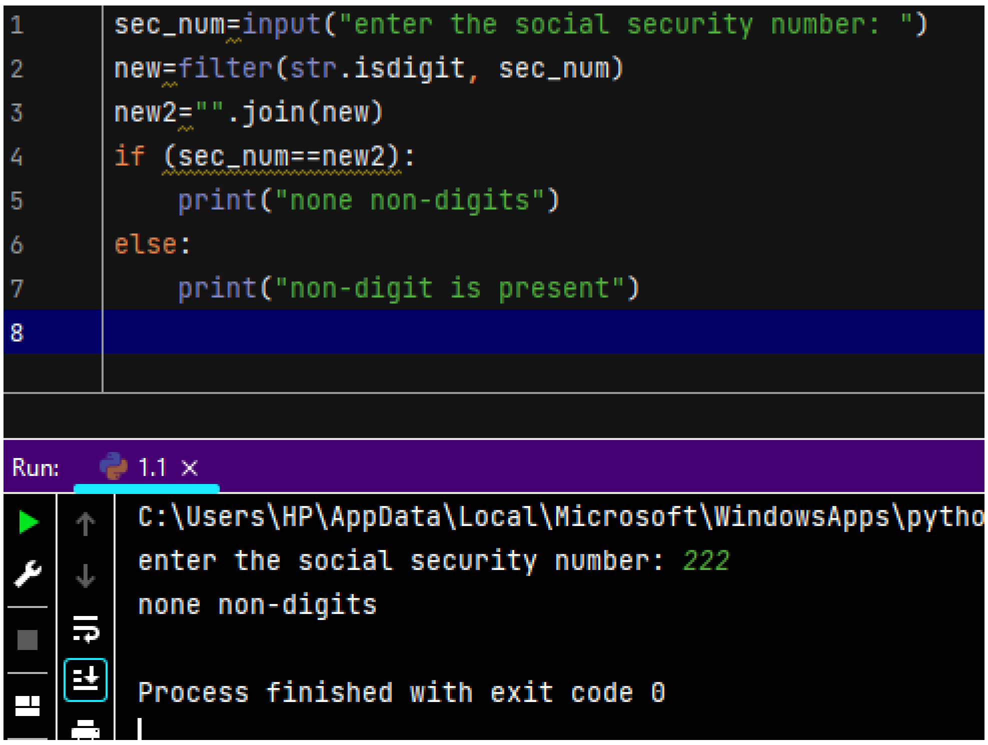


```
fname=input("enter the first name: ")
lname=input("enter the last name: ")
print(lname+","+" "+fname)
```

Run: 1.1

```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\
enter the first name: Jones
enter the last name: William
William, Jones
```

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.

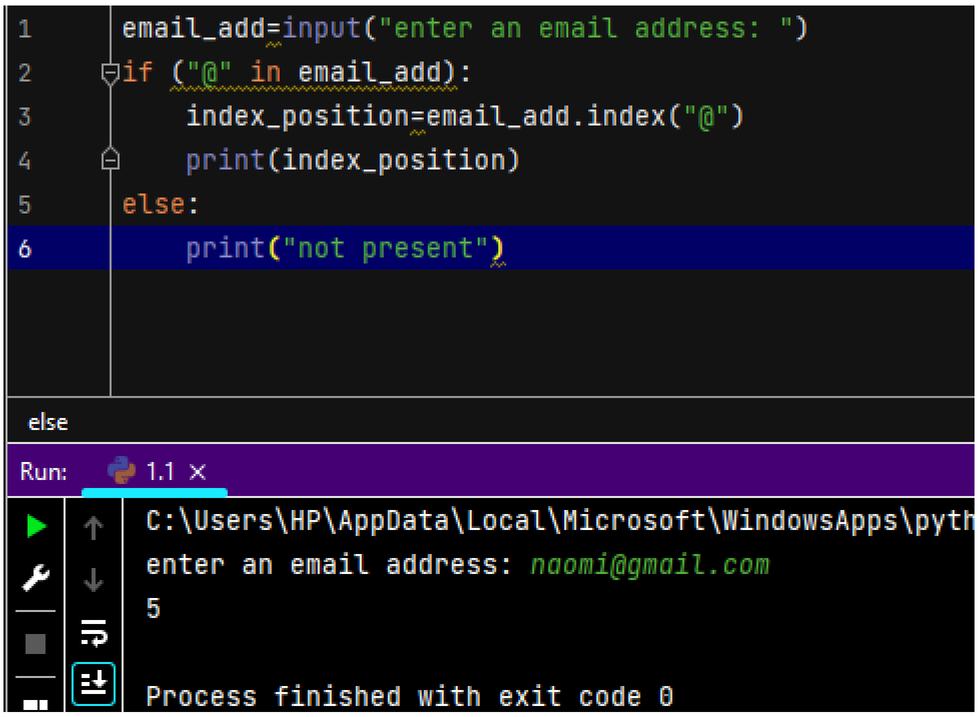


```
sec_num=input("enter the social security number: ")
new=filter(str.isdigit, sec_num)
new2="".join(new)
if (sec_num==new2):
    print("none non-digits")
else:
    print("non-digit is present")
```

Run: 1.1 ×

```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\pytho
enter the social security number: 222
none non-digits

Process finished with exit code 0
```

12. Give an instruction that determines the index of the '@' character in an email address stored in variable `email_addr`.



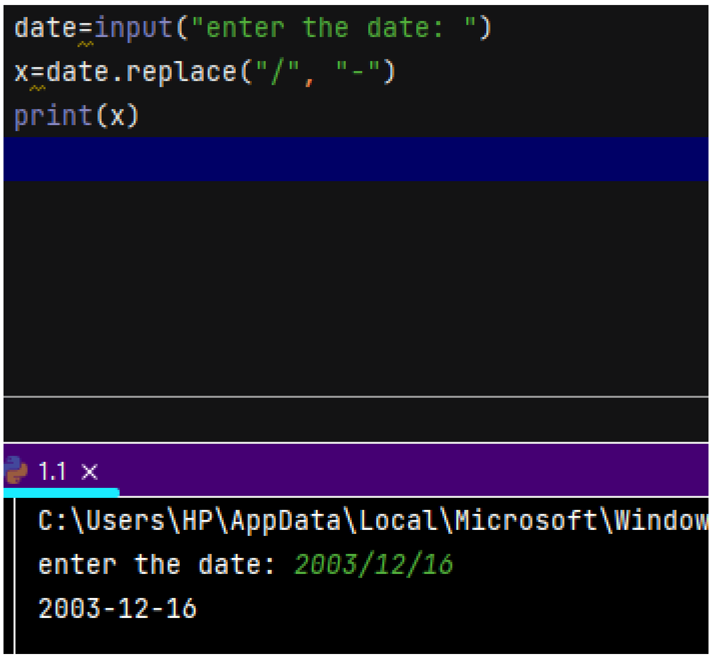
```
email_add=input("enter an email address: ")
if ("@" in email_add):
    index_position=email_add.index("@")
    print(index_position)
else:
    print("not present")
```

else

Run: 1.1 ×

```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\pyth
enter an email address: naomi@gmail.com
5

Process finished with exit code 0
```

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.

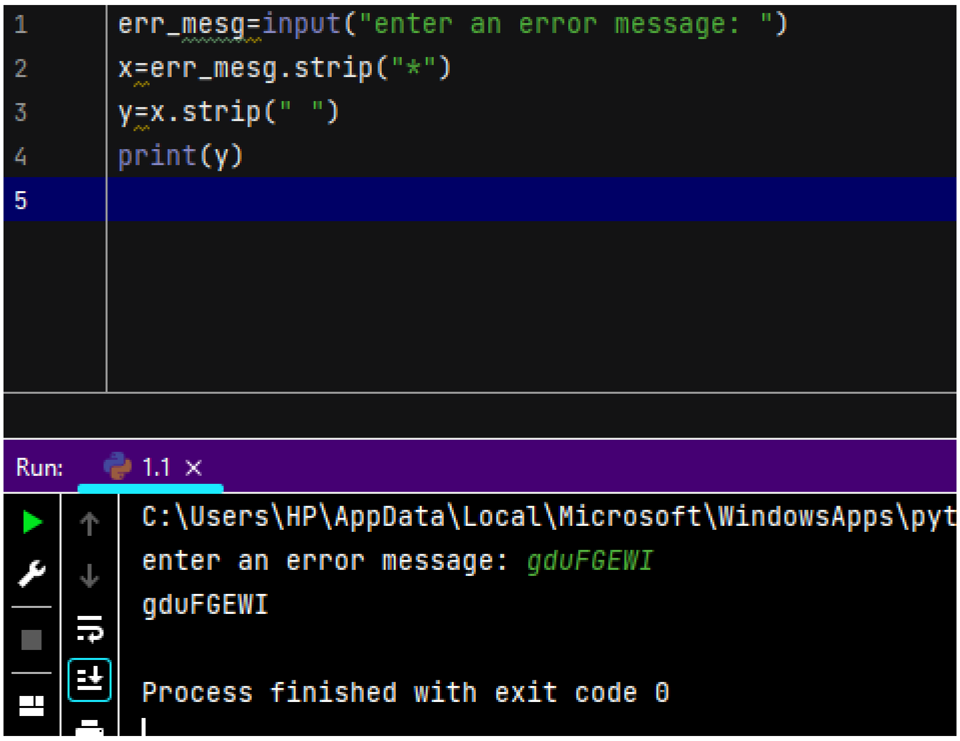


```
date=input("enter the date: ")
x=date.replace("/", "-")
print(x)
```

1.1 ×

```
C:\Users\HP\AppData\Local\Microsoft\Window
enter the date: 2003/12/16
2003-12-16
```

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.



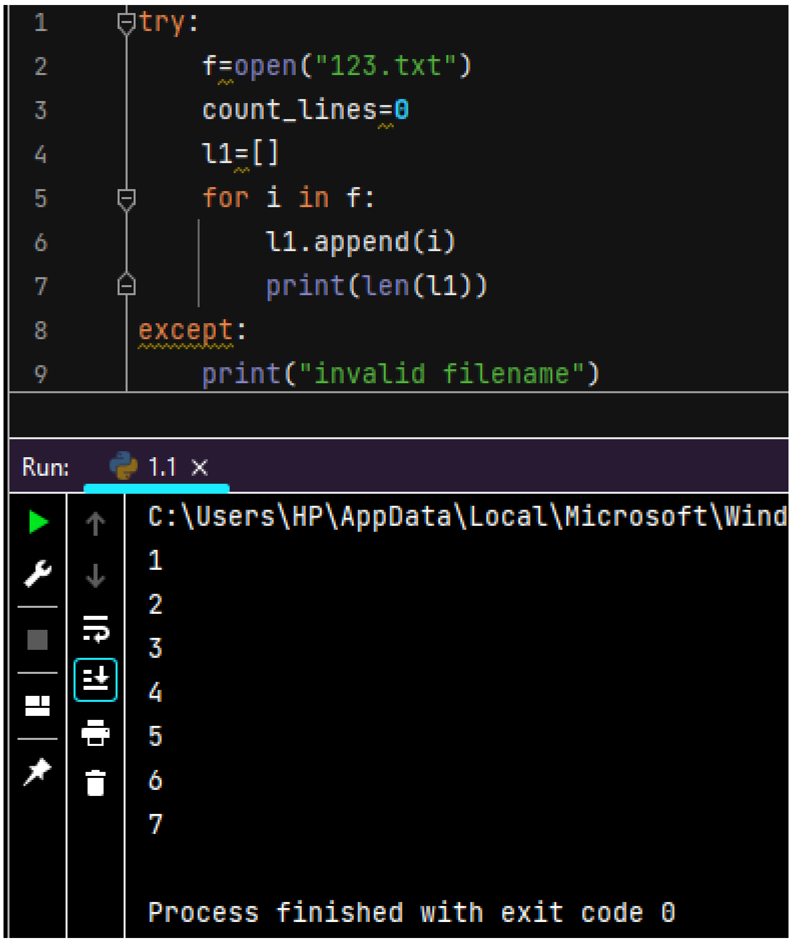
```
err_mesg=input("enter an error message: ")
x=err_mesg.strip("*")
y=x.strip(" ")
print(y)
```

Run: 1.1 ×

```
C:\Users\HP\AppData\Local\Microsoft\WindowsApps\pyt
enter an error message: gduFGEWI
gduFGEWI

Process finished with exit code 0
```

<u>Part 3</u>

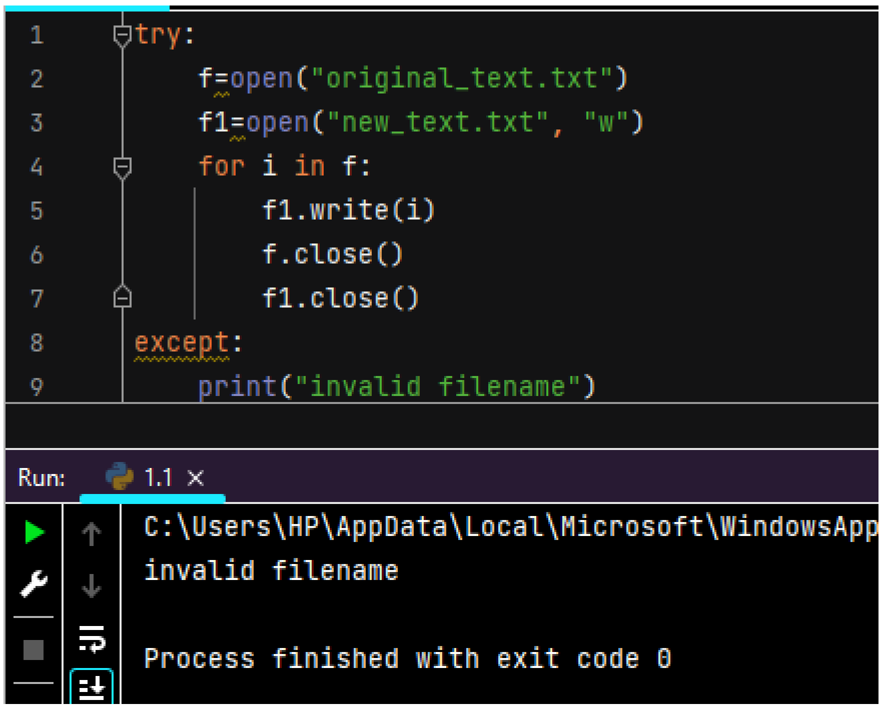
1. Write a program that opens and reads a text file and displays how many lines of text are in the file.



```python
try:
    f=open("123.txt")
    count_lines=0
    l1=[]
    for i in f:
        l1.append(i)
        print(len(l1))
except:
    print("invalid filename")
```

```
C:\Users\HP\AppData\Local\Microsoft\Wind
1
2
3
4
5
6
7

Process finished with exit code 0
```

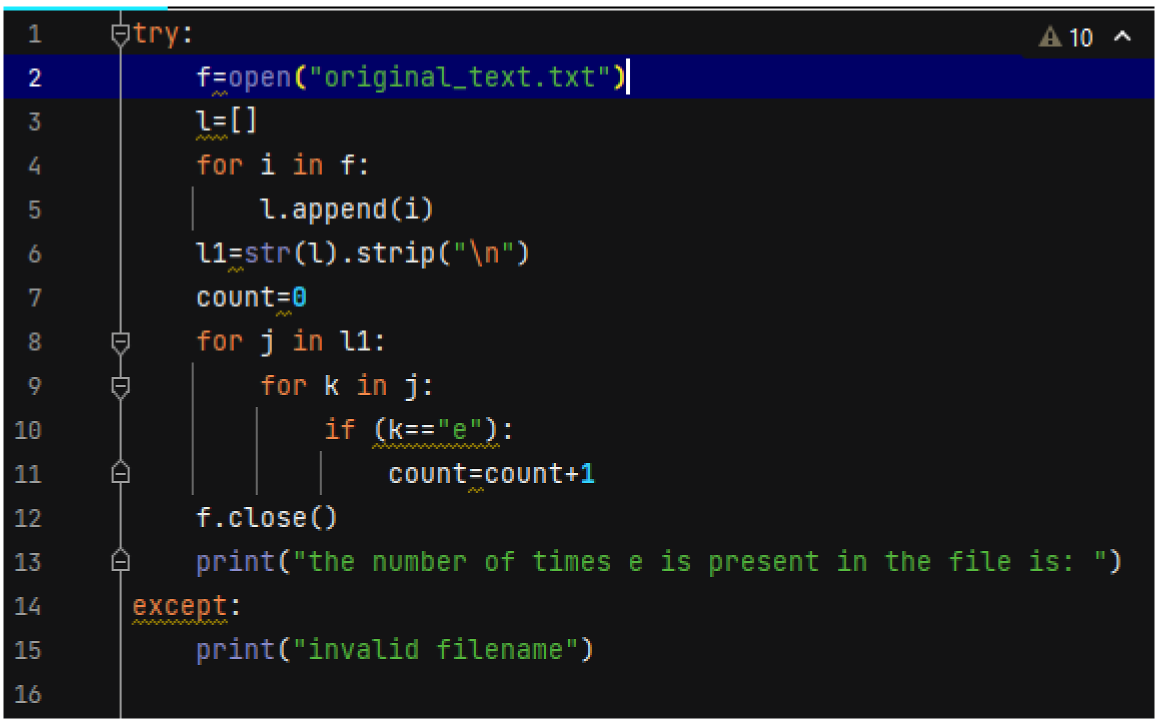
2. Write a program that reads a text file named `original_text`, and writes every other line, starting with the first line, to a new file named `new_text`.



```
try:
    f=open("original_text.txt")
    f1=open("new_text.txt", "w")
    for i in f:
        f1.write(i)
        f.close()
        f1.close()
except:
    print("invalid filename")
```

```
C:\Users\HP\AppData\Local\Microsoft\WindowsApp
invalid filename

Process finished with exit code 0
```

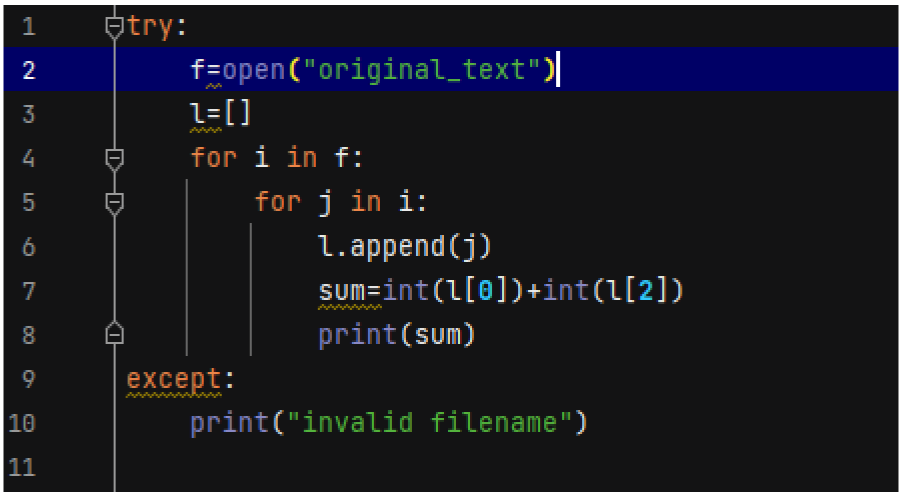
3. Write a program that reads a text file named `original_text`, and counts how many time the letter 'e' occurs (the most frequently occurring letter in English), and displays how many occurrences there are.



```
try:
    f=open("original_text.txt")
    l=[]
    for i in f:
        l.append(i)
    l1=str(l).strip("\n")
    count=0
    for j in l1:
        for k in j:
            if (k=="e"):
                count=count+1
    f.close()
    print("the number of times e is present in the file is: ")
except:
    print("invalid filename")

```

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.`



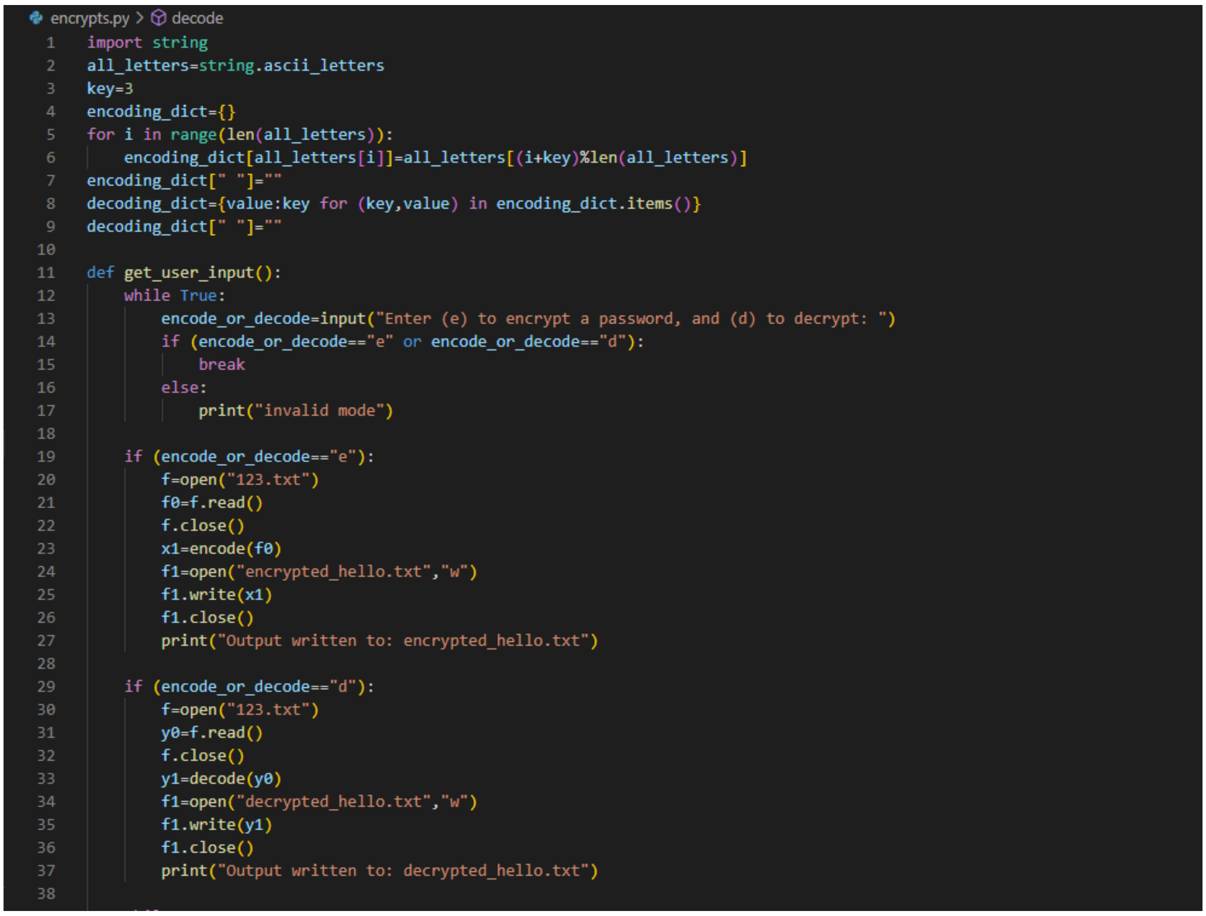
```
try:
    f=open("original_text")
    l=[]
    for i in f:
        for j in i:
            l.append(j)
            sum=int(l[0])+int(l[2])
            print(sum)
except:
    print("invalid filename")
```

Part 4 (Optional)

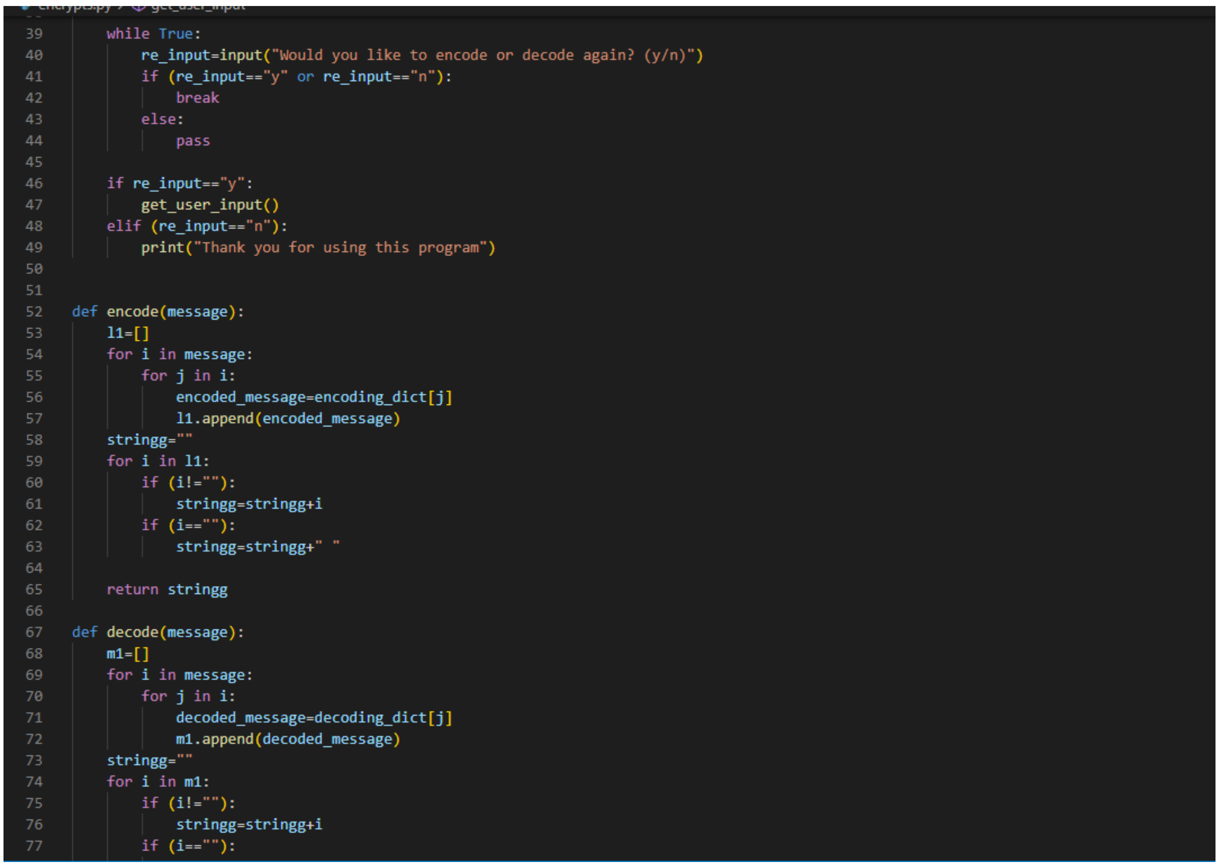
Write a Python program that encrypts and decrypts text files using a substitution cipher. Your program should ask the user for the name of a text file and whether they would like to encrypt or decrypt. Once the process is complete, you should write the output to a new text file with a modified name:

```
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
```

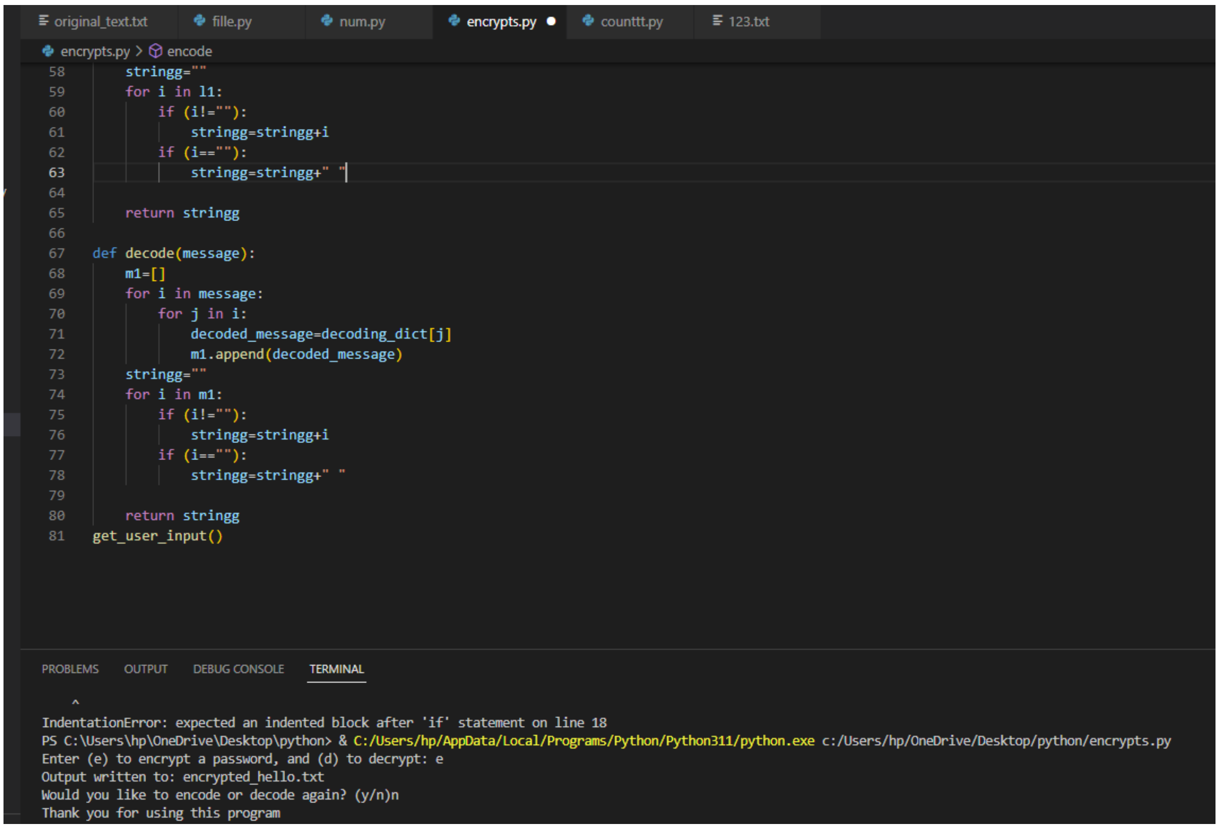
Your program should catch exceptions and print helpful error messages. You should use your solution to Coding Challenge 03 to help you.



```python
import string
all_letters=string.ascii_letters
key=3
encoding_dict={}
for i in range(len(all_letters)):
    encoding_dict[all_letters[i]]=all_letters[(i+key)%len(all_letters)]
encoding_dict[" "]=""
decoding_dict={value:key for (key,value) in encoding_dict.items()}
decoding_dict[" "]=""

def get_user_input():
    while True:
        encode_or_decode=input("Enter (e) to encrypt a password, and (d) to decrypt: ")
        if (encode_or_decode=="e" or encode_or_decode=="d"):
            break
        else:
            print("invalid mode")

    if (encode_or_decode=="e"):
        f=open("123.txt")
        f0=f.read()
        f.close()
        x1=encode(f0)
        f1=open("encrypted_hello.txt","w")
        f1.write(x1)
        f1.close()
        print("Output written to: encrypted_hello.txt")

    if (encode_or_decode=="d"):
        f=open("123.txt")
        y0=f.read()
        f.close()
        y1=decode(y0)
        f1=open("decrypted_hello.txt","w")
        f1.write(y1)
        f1.close()
        print("Output written to: decrypted_hello.txt")

```



```python
    while True:
        re_input=input("Would you like to encode or decode again? (y/n)")
        if (re_input=="y" or re_input=="n"):
            break
        else:
            pass

    if re_input=="y":
        get_user_input()
    elif (re_input=="n"):
        print("Thank you for using this program")


def encode(message):
    l1=[]
    for i in message:
        for j in i:
            encoded_message=encoding_dict[j]
            l1.append(encoded_message)
    stringg=""
    for i in l1:
        if (i!=""):
            stringg=stringg+i
        if (i==""):
            stringg=stringg+" "

    return stringg

def decode(message):
    m1=[]
    for i in message:
        for j in i:
            decoded_message=decoding_dict[j]
            m1.append(decoded_message)
    stringg=""
    for i in m1:
        if (i!=""):
            stringg=stringg+i
        if (i==""):
```



original_text.txt fille.py num.py encrypts.py counttt.py 123.txt

encrypts.py > encode

```python
    stringg=""
    for i in l1:
        if (i!=""):
            stringg=stringg+i
        if (i==""):
            stringg=stringg+" "

    return stringg

def decode(message):
    m1=[]
    for i in message:
        for j in i:
            decoded_message=decoding_dict[j]
            m1.append(decoded_message)
    stringg=""
    for i in m1:
        if (i!=""):
            stringg=stringg+i
        if (i==""):
            stringg=stringg+" "

    return stringg
get_user_input()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
    ^
IndentationError: expected an indented block after 'if' statement on line 18
PS C:\Users\hp\OneDrive\Desktop\python> & C:/Users/hp/AppData/Local/Programs/Python/Python311/python.exe c:/Users/hp/OneDrive/Desktop/python/encrypts.py
Enter (e) to encrypt a password, and (d) to decrypt: e
Output written to: encrypted_hello.txt
Would you like to encode or decode again? (y/n)n
Thank you for using this program
```