EXP 1:

num1 = int(input("Enter the first number: "))

num2 = int(input("Enter the second number: "))

temp = num1

num1 = num2

num2 = temp

if num1 > 0:

print("The first number is positive.")

elif num1 < 0:

print("The first number is negative.")

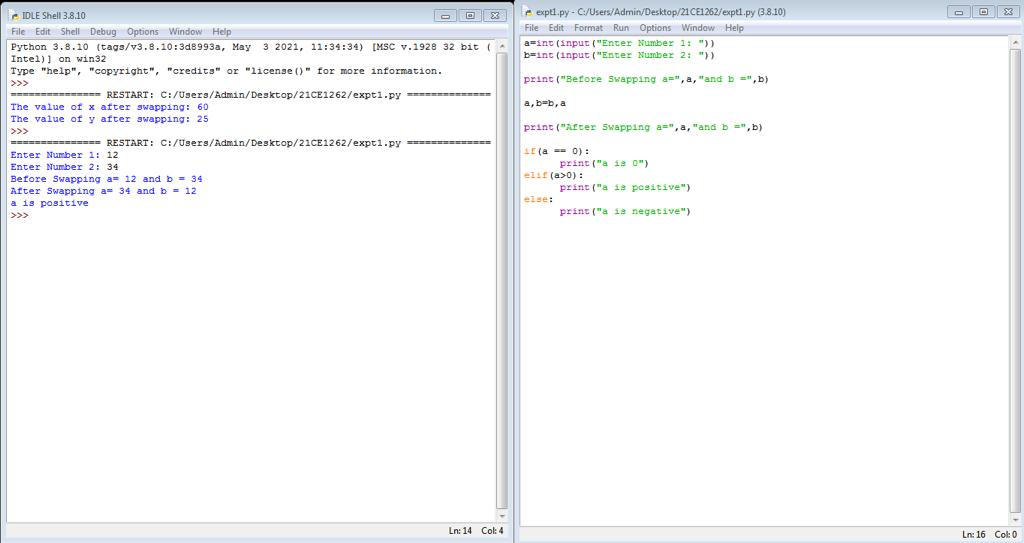
else:

print("The first number is zero.")

print("After swapping , the first number is:", num1)

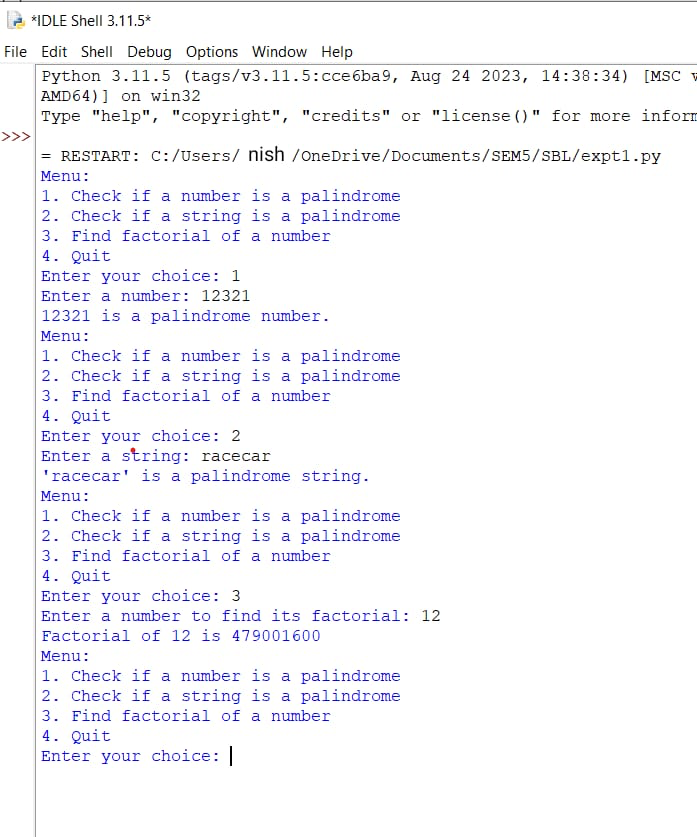
print("After swapping , the first second is:", num2)

OUTPUT:



EXP 2:

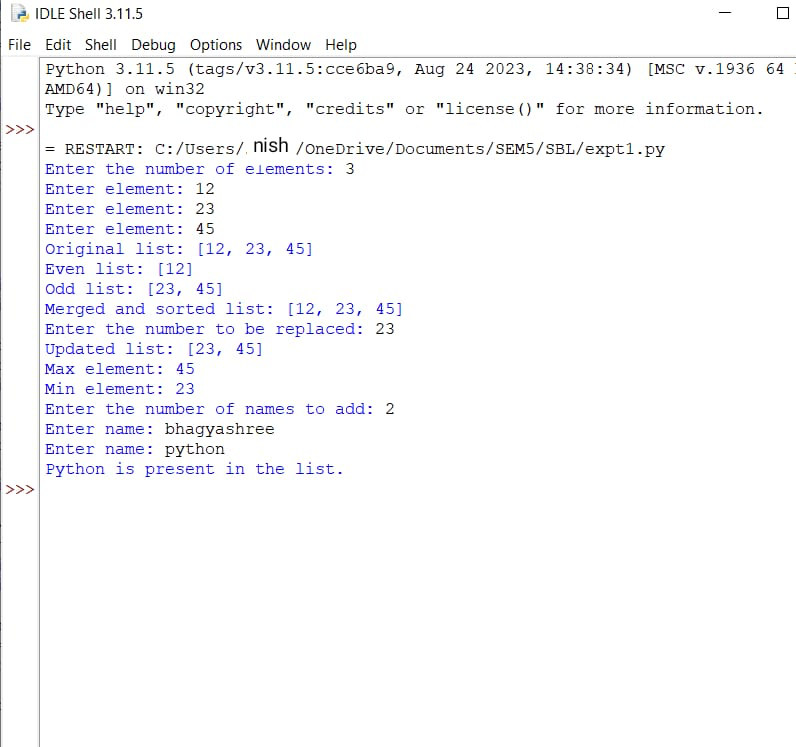
def is\_palindrome\_number(number):  
 num\_str = str(number)  
 return num\_str == num\_str[::-1]  
  
def is\_palindrome\_string(string):  
 cleaned\_string = ''.join(string.split()).lower()  
 return cleaned\_string == cleaned\_string[::-1]  
  
def factorial(n):  
 if n == 0:  
 return 1  
 else:  
 return n \* factorial(n - 1)  
  
while True:  
 print("Menu:")  
 print("1. Check if a number is a palindrome")  
 print("2. Check if a string is a palindrome")  
 print("3. Find factorial of a number")  
 print("4. Quit")  
   
 choice = input("Enter your choice: ")  
  
 if choice == '1':  
 number = int(input("Enter a number: "))  
 if is\_palindrome\_number(number):  
 print(f"{number} is a palindrome number.")  
 else:  
 print(f"{number} is not a palindrome number.")  
  
 elif choice == '2':  
 string = input("Enter a string: ")  
 if is\_palindrome\_string(string):  
 print(f"'{string}' is a palindrome string.")  
 else:  
 print(f"'{string}' is not a palindrome string.")  
  
 elif choice == '3':  
 num = int(input("Enter a number to find its factorial: "))  
 if num < 0:  
 print("Factorial is not defined for negative numbers.")  
 else:  
 result = factorial(num)  
 print(f"Factorial of {num} is {result}")  
  
 elif choice == '4':  
 print("Goodbye!")  
 break  
  
 else:  
 print("Invalid choice. Please select a valid option.")

OUTPUT: 

EXP 3:

def main():  
 lst = []  
 n = int(input("Enter the number of elements: "))  
 for i in range(n):  
 lst.append(int(input("Enter element: ")))  
 print("Original list:", lst)  
  
 even\_lst = []  
 odd\_lst = []  
 for i in lst:  
 if i % 2 == 0:  
 even\_lst.append(i)  
 else:  
 odd\_lst.append(i)  
 print("Even list:", even\_lst)  
 print("Odd list:", odd\_lst)  
  
 merged\_lst = even\_lst + odd\_lst  
 merged\_lst.sort()  
 print("Merged and sorted list:", merged\_lst)  
  
 num1 = int(input("Enter the number to be replaced: "))  
 merged\_lst[0] = num1  
 del merged\_lst[len(merged\_lst) // 2]  
 print("Updated list:", merged\_lst)  
  
 max\_num = max(merged\_lst)  
 min\_num = min(merged\_lst)  
 print("Max element:", max\_num)  
 print("Min element:", min\_num)  
  
 names = []  
 n = int(input("Enter the number of names to add: "))  
 for i in range(n):  
 names.append(input("Enter name: "))  
 merged\_lst += names  
 if "python" in merged\_lst:  
 print("Python is present in the list.")  
 else:  
 print("Python is not present in the list.")  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

OUTPUT:



EXP 4:

from pprint import pprint

t =()

l = []

def menu():

print("\n///MENU///")

print("1. Create & display tuple")

print("2. Display details of a student 'Python'")

print("3. Sort nested tuple")

print("4. Exit")

return (int(input("\nEnter your choice: ")))

def read():

global t

n = int(input("\nEnter the no. of students: "))

for i in range(0,n):

print("\nEnter details for student", i+1, ": ")

name = input("\nEnter Name: ")

rno = input("Enter Roll no: ")

print("Enter the Marks for 5 subjects: ")

marks = [int(x) for x in input().split()]

t1 = (rno, name, marks)

l.append(t1)

t = tuple(l)

print("\nDisplaying the student details:-")

for i in range(0,n):

print("\nRoll no: ", t[i][0])

print("Name: ", t[i][1])

print("Marks: ", t[i][2])

print("\n------------------------------")

return n

def find(n):

flag = 0

for x in t:

if x[1]=='Python':

print("\nThe name 'Python' found in tuple")

print(x)

flag = 1

break

if flag==0:

print("\nThe name 'Python' not found")

def sort():

sortedt = sorted(t, key = lambda x : x[1])

print("\nThe sorted tuple is:-\n")

pprint(sortedt)

c=0

while(c!=4):

c=menu()

if c==1:

n=read()

elif c==2:

find(n)

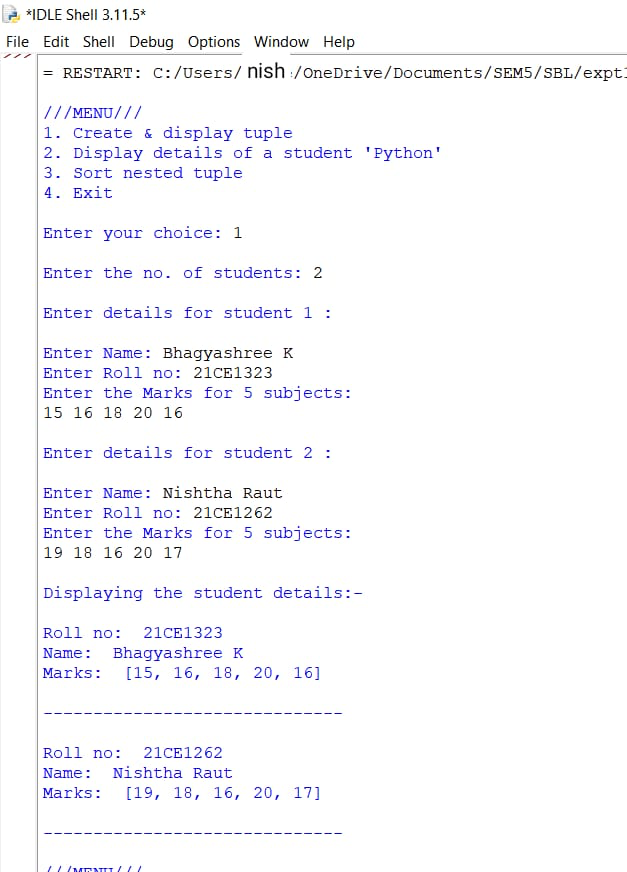
elif c==3:

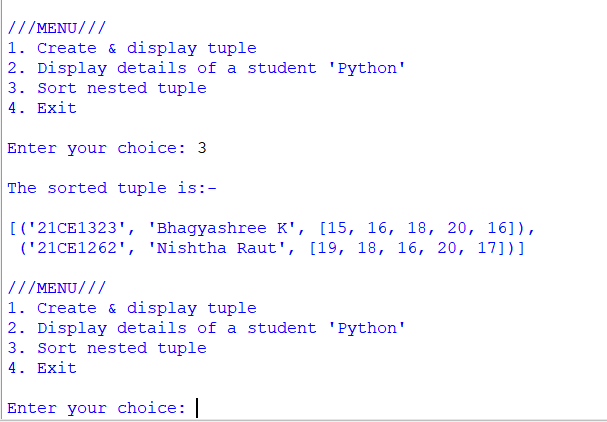
sort()

elif c!=4:

print("\nIncorrect choice")

OUTPUT:





EXP 5:

class Dog:

# Class Variable

animal = 'dog'

# The init method or constructor

def \_\_init\_\_(self, breed, color):

# Instance Variable

self.breed = breed

self.color = color

# Objects of Dog class

Rodger = Dog("Pug", "brown")

Buzo = Dog("Bulldog", "black")

print('Rodger details:')

print('Rodger is a', Rodger.animal)

print('Breed: ', Rodger.breed)

print('Color: ', Rodger.color)

print('\nBuzo details:')

print('Buzo is a', Buzo.animal)

print('Breed: ', Buzo.breed)

print('Color: ', Buzo.color)

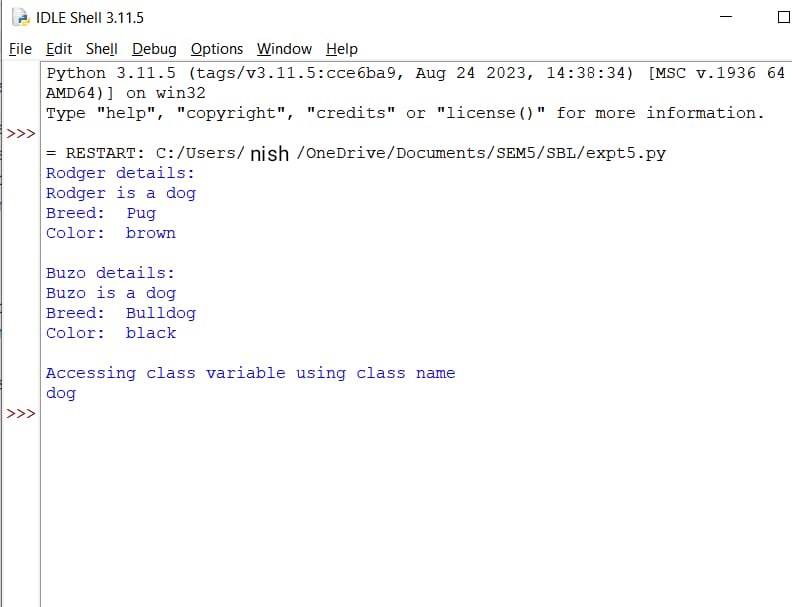
# Class variables can be accessed using class

# name also

print("\nAccessing class variable using class name")

print(Dog.animal)

OUTPUT:



EXP 6:

class Person:

def \_\_init\_\_(self, id, name):

self.ID = id

self.name = name

def display(self):

print("\nID = ", self.ID)

print("\nName = ", self.name)

class Student(Person):

def \_\_init\_\_(self, x, y, id, name):

super().\_\_init\_\_(id, name)

#allows the child class to access the parent class's init() property.

self.sub1 = x

self.sub2 = y

def display(self):

print("\nStudent ID = ", self.ID)

print("Student Name = ", self.name)

print("Marks in subject 1 = ", self.sub1)

print("Marks in subject 2 = ", self.sub2)

class Sports:

def showsports(self, s=None):

self.sportsmarks = s

if s!=None:

print("\nSports marks = ", self.sportsmarks)

class Result(Student, Sports):

def \_\_init\_\_(self, id, name, x, y):

super().\_\_init\_\_(x, y, id, name)

def total(self):

if self.sportsmarks is not None:

sum1 = self.sub1 + self.sub2 + self.sportsmarks

else:

sum1 = self.sub1 + self.sub2

print("\nWe have, Total Marks = ", sum1, "\n")

print("Enter the student details:-")

id = input("\nEnter Student ID: ")

name = input("\nEnter Student Name: ")

m1 = int(input("\nEnter marks in first subject: "))

m2 = int(input("\nEnter marks in second subject: "))

obj = Result(id, name, m1, m2)

obj.display()

c = input("\nDo you want to enter sports marks? (y/n): ")

if c=='y':

sm = int(input("Enter sports marks: "))

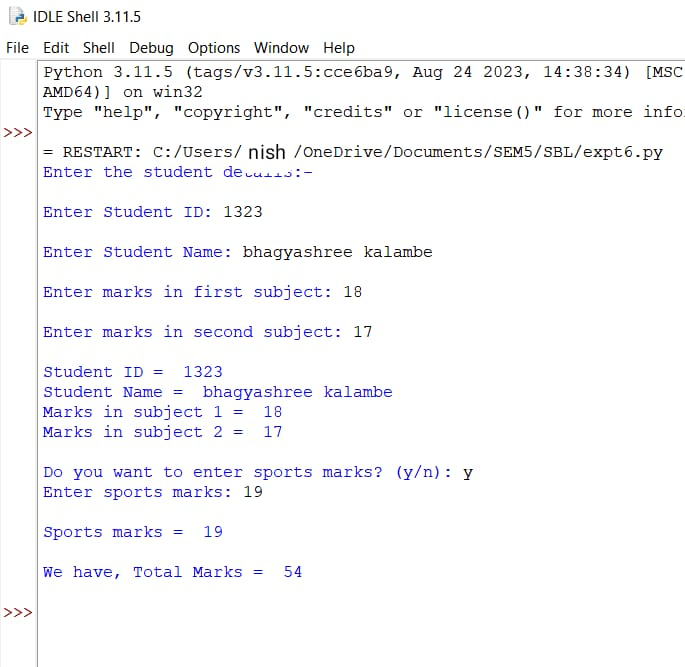
obj.showsports(sm)

else:

obj.showsports()

obj.total()

OUTPUT:



EXP 7:

import sys

randomList = ['a', 0, 4]

for entry in randomList:

try:

print("The entry is", entry)

r = 1/int(entry)

break

except:

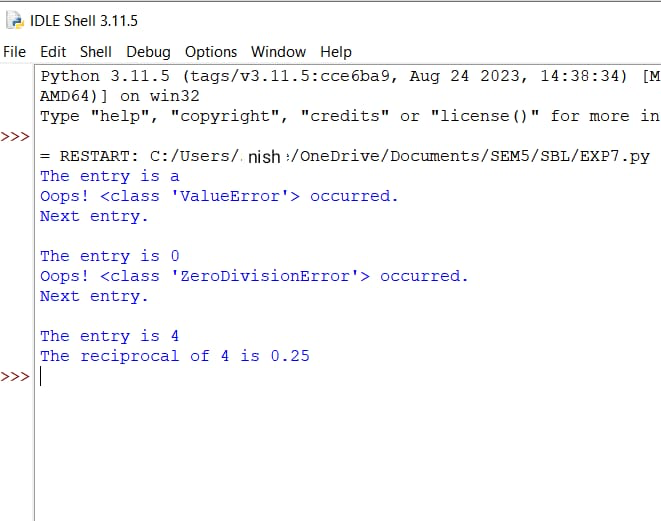
print("Oops!", sys.exc\_info()[0], "occurred.")

print("Next entry.")

print()

print("The reciprocal of", entry, "is", r)

OUTPUT:



EXPT 8:

A)input\_file\_name = input("Enter the name of the input file: ")

# Output file name

output\_file\_name = input("Enter the name of the output file: ")

try:

    # Open the input file for reading

  with open(input\_file\_name, 'r') as input\_file:

        # Read the content of the input file

        content = input\_file.read()

  # Open the output file for writing

  with open(output\_file\_name, 'w') as output\_file:

        # Write the content to the output file

        output\_file.write(content)

  print("Content copied successfully.")

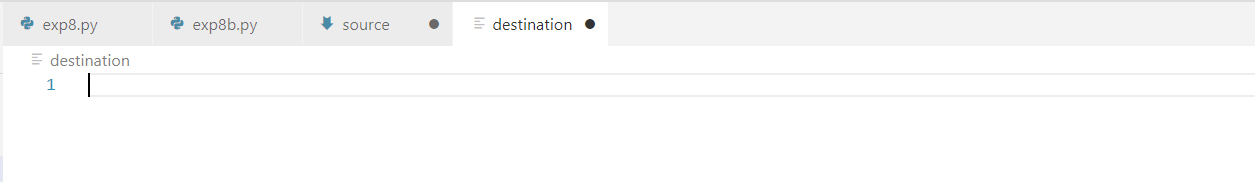
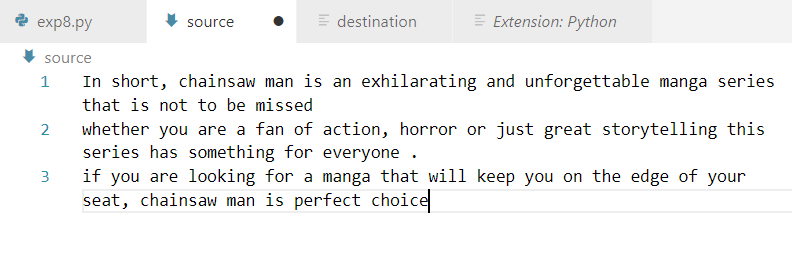
except FileNotFoundError:

    print("File not found.")

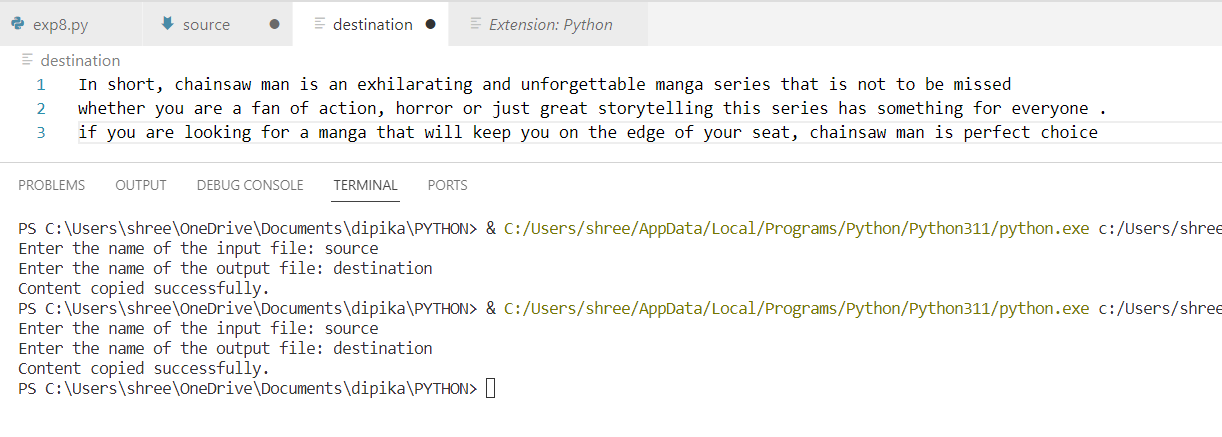
except Exception as e:

    print("An error occurred:", str(e))

OUTPUT:

BEFORE:

After



EXP8:

B)

file\_name = input("Enter the name of the file: ")

try:

    # Open the file for appending

   with open(file\_name, 'a') as file:

        # Input data to append

        data\_to\_append = input("Enter data to append: ")

        # Append the data to the file

        file.write(data\_to\_append + "\n")

    # Open the file for reading and display its entire content

   with open(file\_name, 'r') as file:

        content = file.read()

        print("Updated content of the file:")

        print(content)

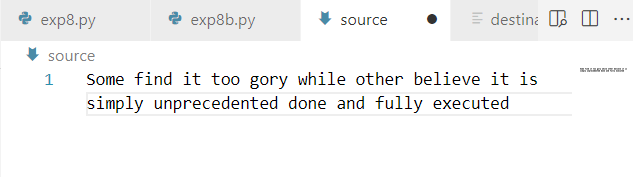
except FileNotFoundError:

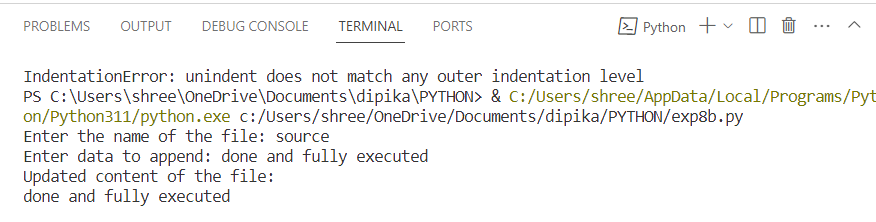
    print("File not found.")

except Exception as e:

    print("An error occurred:", str(e))

OUTPUT:





EXP9:A)

file\_name = input("Enter the name of the file: ")

try:

    with open(file\_name, 'r') as file:

        # Read the content of the file

        content = file.read()

        # Count the number of lines, words, and characters

        lines = content.split('\n')

        num\_lines = len(lines)

        words = content.split()

        num\_words = len(words)

        num\_chars = len(content)

    print("Number of lines:", num\_lines)

    print("Number of words:", num\_words)

    print("Number of characters:", num\_chars)

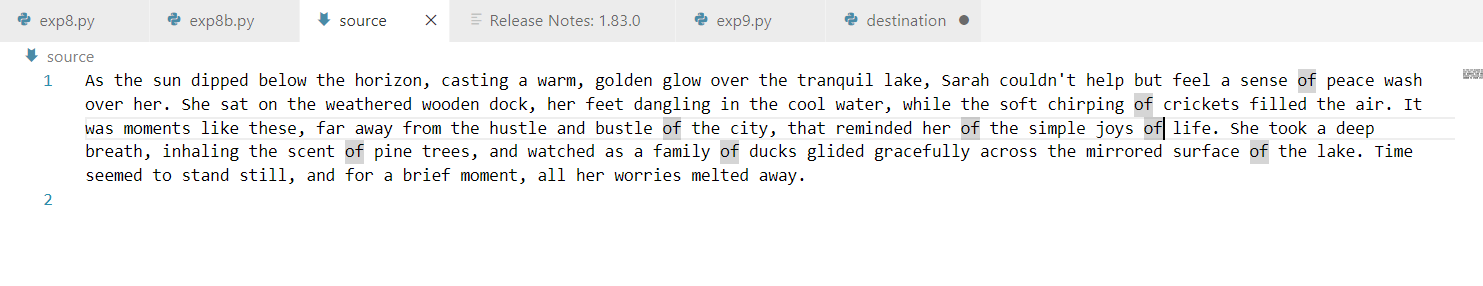
except FileNotFoundError:

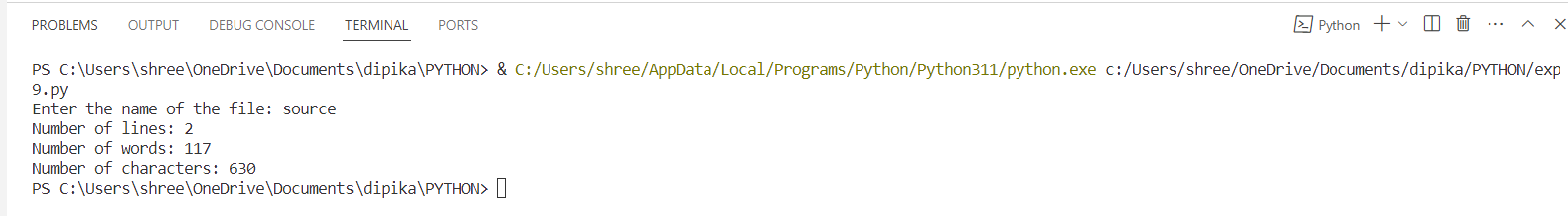
    print("File not found.")

except Exception as e:

    print("An error occurred:", str(e))

OUTPUT:





EXP9: B)

import os

# Get the list of files in the current directory

files = os.listdir()

print("Files in the current directory:")

for file in files:

    if os.path.isfile(file):

        print(file)

OUTPUT:

