1. Implement a Python class MaxHeap that supports the following operations: insert,

delete, and get\_max. Ensure the operations maintain the properties of a max-heap.

class MaxHeap:

def \_\_init\_\_(self):

self.heap = []

def insert(self, val):

self.heap.append(val)

self.\_sift\_up(len(self.heap) - 1)

def delete(self):

if len(self.heap) == 0:

return None

if len(self.heap) == 1:

return self.heap.pop()

root = self.heap[0]

self.heap[0] = self.heap.pop()

self.\_sift\_down(0)

return root

def get\_max(self):

if len(self.heap) == 0:

return None

return self.heap[0]

def \_sift\_up(self, idx):

parent = (idx - 1) // 2

if idx > 0 and self.heap[idx] > self.heap[parent]:

self.heap[idx], self.heap[parent] = self.heap[parent], self.heap[idx]

self.\_sift\_up(parent)

def \_sift\_down(self, idx):

left = 2 \* idx + 1

right = 2 \* idx + 2

largest = idx

if left < len(self.heap) and self.heap[left] > self.heap[largest]:

largest = left

if right < len(self.heap) and self.heap[right] > self.heap[largest]:

largest = right

if largest != idx:

self.heap[idx], self.heap[largest] = self.heap[largest], self.heap[idx]

self.\_sift\_down(largest)

heap = MaxHeap()

heap.insert(50)

heap.insert(28)

heap.insert(51)

heap.insert(37)

heap.insert(49)

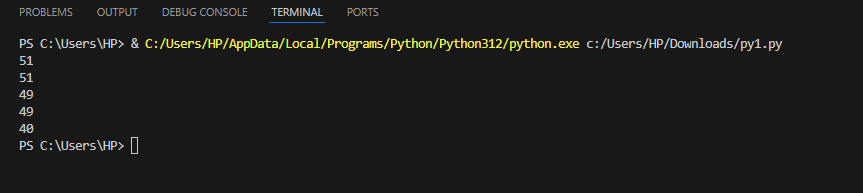
print(heap.get\_max())

print(heap.delete())

print(heap.get\_max())

print(heap.delete())

print(heap.get\_max())



5. Write a Python function to compute the nth Fibonacci number using recursion.

def fibonacci(n):

if n <= 0:

return 0

elif n == 1:

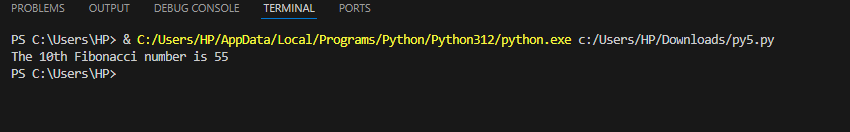
return 1

else:

return fibonacci(n - 1) + fibonacci(n - 2)

n = 10

print(f"The {n}th Fibonacci number is {fibonacci(n)}")



6. Write a Python function that divides two numbers and handles the case where the divisor

is zero by returning a custom error message.

def divide\_numbers(numerator, divisor):

try:

result = numerator / divisor

return result

except ZeroDivisionError:

return "Error: Division by zero is not allowed."

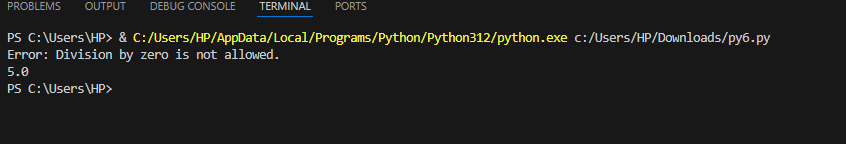
numerator = 10

divisor = 0

print(divide\_numbers(numerator, divisor))

divisor = 2

print(divide\_numbers(numerator, divisor))



8. Write a Python function that takes two numbers and an operator (as a string) and

performs the corresponding arithmetic operation (addition, subtraction, multiplication, or

division).

def calculate(num1, num2, operator):

if operator == '+':

return num1 + num2

elif operator == '-':

return num1 - num2

elif operator == '\*':

return num1 \* num2

elif operator == '/':

return num1 / num2

num1 = 10

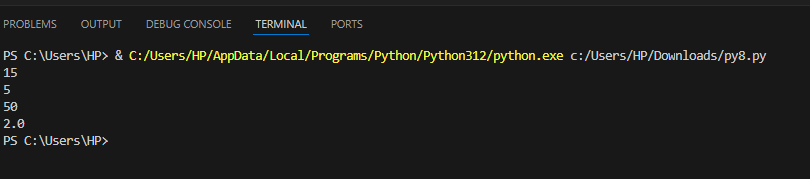
num2 = 5

print(calculate(num1, num2, '+'))

print(calculate(num1, num2, '-'))

print(calculate(num1, num2, '\*'))

print(calculate(num1, num2, '/'))



9. Write a Python function that generates a random password. The password should

contain a mix of uppercase letters, lowercase letters, digits, and special characters.

import random

import string

def generate\_password(length=12):

if length < 4:

return "Error: Password length should be at least 4."

uppercase = string.ascii\_uppercase

lowercase = string.ascii\_lowercase

digits = string.digits

special\_characters = string.punctuation

password = [ random.choice(uppercase),random.choice(lowercase),random.choice(digits),

random.choice(special\_characters)]

all\_characters = uppercase + lowercase + digits + special\_characters

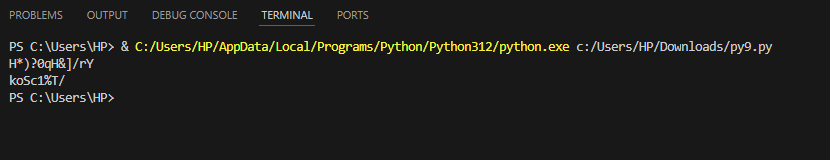
password += random.choices(all\_characters, k=length-4)

random.shuffle(password)

return ''.join(password)

print(generate\_password(12))

print(generate\_password(8))



10. Write a Python function that takes a 2D list (matrix) and returns its transpose.

a = int(input("Enter the number of rows: "))

b = int(input("Enter the number of columns: "))

m1 = []

for i in range(a):

arr = []

for j in range(b):

m = int(input(f"Enter element [{i+1}, {j+1}]: "))

arr.append(m)

m1.append(arr)

print("Original matrix:")

for i in range(a):

for j in range(b):

print(m1[i][j], end=" ")

print()

print("Transpose of the matrix:")

for j in range(b):

for i in range(a):

print(m1[i][j], end=" ")

print()

