4 a) Develop a Python program that reads a text file and prints words of specified lengths (e.g.,

three, four, five, etc.) found within the file.

f = open("demofile.txt", "r")

paragraph = f.read()

#line = "Hello, this is an example line."

words = paragraph.split()

print(words)

print(type(words))

def count\_word\_lengths(words):

    # Create a dictionary to store word lengths and their counts

    word\_lengths = {}

    for word in words:

        length = len(word)

        if length in word\_lengths:

            word\_lengths[length] += 1

        else:

            word\_lengths[length] = 1

    return word\_lengths

word\_lengths = count\_word\_lengths(words)

print("Word lengths and their counts:")

for length, count in word\_lengths.items():

    print(f"Length {length}: {count} words")

for length in word\_lengths:

    # Filter words of the current length

    words\_of\_length = [word for word in words if len(word) == length]

    print(f"Length {length}: {', '.join(words\_of\_length)}")

b) Write a python code to take a csv file as input with coordinates of points in three dimensions. Find out the two closest points.

import csv

import math

def read\_csv(filename):

    points = []

    with open(filename, 'r') as file:

        reader = csv.reader(file)

        next(reader)  # Skip header row if there is one

        for row in reader:

            try:

                points.append([float(coord) for coord in row])

            except ValueError:

                print(f"Skipping row: {row} - Could not convert to float.")

    return points

def distance(p1, p2):

    return math.sqrt((p1[0] - p2[0])\*\*2 + (p1[1] - p2[1])\*\*2 + (p1[2] - p2[2])\*\*2)

def find\_closest\_points(points):

    min\_distance = float('inf')

    closest\_pair = None

    for i in range(len(points)):

        for j in range(i + 1, len(points)):

            dist = distance(points[i], points[j])

            if dist < min\_distance:

                min\_distance = dist

                closest\_pair = (points[i], points[j])

    return closest\_pair, min\_distance

if \_\_name\_\_ == "\_\_main\_\_":

    filename = 'Coordinate.csv'

    points = read\_csv(filename)

    closest\_pair, min\_distance = find\_closest\_points(points)

    print("The closest points are:", closest\_pair)

    print("The distance between them is:", min\_distance)