1 Database Assembly Code

```
################################################################################
#
#     3D Printing Research Ingest
#     Programmed by Evan Grubis, 2021
#     ### Denotes Comments for the software
#     # Denotes commented sections for various reasons
################################################################################

# INSTRUCTIONS:
# Download the "Data Collection" folder on Box using the three dots (...)
# Unzip the file using your operating system
# Run the program and select the unzipped [Datapath] folder

outputfilename = 'data.csv'

################################################################################
#
#     Import required libraries
#
################################################################################

import os
import re
import glob
import csv
import pandas as pd
import tkinter as tk
from tkinter import filedialog


################################################################################
###
###     Define functions
###
################################################################################

### Asks the user to select the root file path to the data
def getrootfilepath():
    downloads = os.getcwd()                         # finds the downloads folder
    downloads = downloads.split("\\")
    downloads = downloads[0] + "/" + downloads[1] + "/" + downloads[2] + "/Downloads"
    root = tk.Tk()
    root.withdraw()
    root_path = filedialog.askdirectory(title='Please␣select␣the␣unzipped␣[Data␣Collection]␣folder',
                                        initialdir=downloads)    # creates dialogue window
    return root_path                                             # returns the selected folder

### Reads all the filenames in a given directory
def get_files(directory):
    filenames = glob.glob(directory + "/0␣degree␣orientation/*/*.csv")
```

```
        # searches 0 degree folder
    filenames = filenames + glob.glob(directory + "/90_degree_orientation/*/*.csv")
        # searches 90 degree folder
    print(str(len(filenames)) + "_.CSV_files_discovered")
        # outputs how many CSV files were found
    return filenames



### Splits, cleans, and extracts useful info from a filepath
def get_info(infopath):
    returnme = []
        # array to return
    infopath = re.split('[.]', infopath)[0]
        # gets rid of the '.csv'
    infopath = infopath.split('\\')
        # swap \ for /
    tensilepath = infopath[0] + "/" + infopath[1]
        # create the tensile data path for later
    end = infopath[2]
    infopath = infopath + end.split('_')
        # split at underscores
    cleanpath = infopath[0] + '/' + infopath[1] + '/' + infopath[2] + '.csv'
        # reconstructs filepath
    data = pd.read_csv(cleanpath, sep=',',header=None).values
        # reads value from filepath
    infopath = infopath[3:]
    #   0               1      2      3          4               5          6
    #  ReplicateNumber_ Length_Width_LayerHeight_NozzleDiameter_PrintSpeed_Orientation

    #
    #
    # Orientation LH ND PS AT FS MF M TS Run Sample


    i = 1                                                   # init variable to keep track of sample
    tensiledata = glob.glob(tensilepath + "/*/*.csv")       # reads the tensile data
    if len(tensiledata) > 0:                                # if the tensile data exists
        tensiledata = tensiledata[0].split('\\')                                    # swap \ to /
        tensiledata = tensiledata[0] + "/" + tensiledata[1] + "/" + tensiledata[2]
        tensdata = pd.read_csv(tensiledata, sep=',', header=None, skiprows=6).values # read tensile data
    else:
        tensdata = []
                # if the tensile data doesn't exist, make empty array
    for array in data:                          # run through all the data from the original CSV file
        if len(tensdata) > 0:                   # if tensile data exists
            if int(infopath[0]) == 1:              # if Run = 1
                ten = tensdata[1 + i][12]       # set temp variables to correct data
                mom = tensdata[1 + i][5]
            elif int(infopath[0]) == 2:            # if Run = 2
                ten = tensdata[5 + i][12]       # set temp variables to correct data
                mom = tensdata[5 + i][5]
            elif int(infopath[0]) == 3:            # if Run = 2
                ten = tensdata[9 + i][12]       # set temp variables to correct data
                mom = tensdata[9 + i][5]
            else:                               # if Run != 1 or 2
                ten = None                      # set temp variables to null
                mom = None
        else:                                   # if tensile data does not exist
            ten = 'n/d'                         # set temp variables to "n/d"
            mom = 'n/d'
        if len(infopath) == 8:                     # if the data has comments
            row = [infopath[6], float(infopath[3]) / 10, float(infopath[4]) / 10, infopath[5], array[0], array[1],
                   array[2], mom, ten, infopath[0], i, infopath[7], cleanpath]  # set the data as needed+
        else:                                   # if the data does not have comments
```

```
            row = [infopath[6], float(infopath[3]) / 10, float(infopath[4]) / 10, infopath[5], array[0], array[1],
                   array[2], mom, ten, infopath[0], i, "", cleanpath]   # set the data as needed
            returnme.append(row)                        # append the data to the return array
            i += 1                                       # increment the sample number
        return returnme                                  # once all data has been read, return the array


################################################################################
###
###     Begin Main Loop
###
################################################################################

if __name__ == "__main__":
    print("Initiated root directory popup")            # prints some information to aid with debugging
    rootpath = getrootfilepath()                        # triggers the filepath popup function
    print("Root directory assignment successful")
    print("The root path is " + rootpath)               # prints root filepath

    big_array = [['Orientation [deg]', 'Layer Height [mm]', 'Nozzle Diameter [mm]', 'Print Speed [mm/min]',
                  'Avg Temperature [C]', 'Avg Filament Speed [mm/s]', 'Avg Force [g]', 'Modulus [MPa]',
                  'Tensile Stress at Tensile Strength [MPa]', 'Run', 'Sample', 'Notes', 'Filepath']]
                                            # manually sets the first row of the CSV file with the titles
    paths = get_files(rootpath)
        # triggers the get_files function with the rootpath

    for filename in paths:                              # prints all extracted info in a readable format
        try:                                            # try element to elegantly catch any errors in formatting
            info = get_info(filename)                   # get the info from the filename
        except Exception as e:                          # catches said exceptions
            print(filename + " could not be read")      # prints helpful information
            print(e)
            continue                                    # moves on to next filename
        big_array += info                               # adds the info to the final file

    lengthBig = len(big_array) - 1                      # finds the length of the array without the first row
                                                            # (which we added manually)
    if lengthBig <= 0:
        print("Zero entries were found. Ensure you are using the correct folder. If this was an unzipped folder, "
              "you may have to go down another level to get to the correct folder")
                                                        # prints an error if no info has been added at this point
    else:
        print(str(lengthBig) + " sample entries found") # prints how many samples were found
    try:
        with open(outputfilename, "w") as my_csv:                       # write CSV file
            csvWriter = csv.writer(my_csv, lineterminator = '\n')
            csvWriter.writerows(big_array)
            print("File '" + outputfilename + "' created successfully")
        launchExcel = 1
    except:
        print("ERROR: Make sure you closed the CSV file before running the program. The file was not written")
        launchExcel = 0

    if launchExcel == 1:
        try:                                                            # launch excel with the output file open
            os.system("start EXCEL.EXE " + outputfilename)
            print(outputfilename + " opened in Excel successfully")
        except:
            print("Excel was unable to be launched")
```