**Programming Quiz**

This problem sheet contains 2 problems

**Problem 1**

data.csv contains a fictitious dataset of male sample populations of a rare species of lifeform from planets. It is assumed that there are only 2 sexes, male and female.

You should process this file into a suitable format so that it can be used in a web application.

Create a single page web application in a technology of your choice.

The web application must run locally (local host) and any instructions for starting the application should be documented clearly.

*Use Cases*

1. A user wishes to see to the total number of Males or total number of Females across each of the planets. It is important for the user to see all planet counts at the same time so they can easily compare one planet to another.
2. The user wants to see the planets in alphabetic ascending order.
3. The user wants to be able to change whether they are looking at Males, Females or both Males and Females. If they are looking at both Males and Females then they want to see the number of Females and the numbers of Males (not the combined total).
4. The user also wants to know the total population of Males and Females across all planets and they would also like to see this displayed on the page. They would like this to be linked to the functionality in 2) so for example if they are looking at Males then the total number of Males will be displayed. In the case of both Males and Females selected then the user expects the total number of Males + total number of Females to be displayed.
5. The user wants to perform an arbitrary calculation called ‘Population Scaling’ which is (a \* x)/(b-1)^2 , (where x is the result computed in 4), a and b are floating point numbers in the range [2,10000]. The user expects to be able to enter values for a and b in the interface and click a button to see the result.

**Problem 2**

Copy the snippet of code below into a Python 3 environment and run it. You will need to have Pandas installed as a dependency.

"""

Task:

The peak memory usage of this script is too high and runs out of memory on users machines.

How can you fix the memory usage without breaking the test?

You should be able to relatively easily reduce the memory usage by \_at least\_ 25%

"""

from pandas.util import hash\_pandas\_object

import pandas as pd

import hashlib

import typing

import random

# Deterministic randomness for the assert at the end

random.seed(42)

TRANSFORMS = 10

ROWS = 20000000

DATA = [random.random() for \_ in range(ROWS)]

def update\_df(df\_orig: pd.DataFrame, df\_new: pd.DataFrame) -> pd.DataFrame:

    return pd.concat((df\_orig, df\_new), axis=1)

class Transform:

    """A transform that adds a column of random data"""

    def \_\_init\_\_(self, var: str):

        self.var = var

    def transform(self, df: pd.DataFrame) -> pd.DataFrame:

        df\_new = pd.DataFrame({self.var: DATA})

        return update\_df(df, df\_new)

class Pipeline:

    def \_\_init\_\_(self):

        self.df = pd.DataFrame()

        self.transforms = [

            Transform(f'v{i}') for i in range(TRANSFORMS)

        ]

    def run(self):

        for t in self.transforms:

            self.df = t.transform(self.df)

        return self.df

if \_\_name\_\_ == '\_\_main\_\_':

    pipe = Pipeline()

    df = pipe.run()

    # Dont break the test

    assert hashlib.sha256(pd.util.hash\_pandas\_object(df, index=True).values).hexdigest() == '867567dc7d46f77af2bca9804ac366a5165d27612de100461b699bd23094ab90'