1. **Before they switch all the sensors back on they want to have a solution in place. However, right now there is no data for you to use. You have been asked to generate a dummy dataset that you can use to prove your solutions work with.**

```
import pandas as pd
import numpy as np
idx = np.arange(0,16)
cols = np.arange(1,33)
f = pd.DataFrame(data = np.random.uniform(size = (16,32)), columns = cols, index = idx)
print(f)
```

2.1 For an extra challenge you can try to date and time stamp each interval of data collection

```
import pandas as pd
import numpy as np
DSI = np.arange(0,16)
readings = np.arange(1,33)

for x in range(16) :
   Px = pd.DataFrame(data = np.random.uniform(size = (16,32)), columns = readings, index = DSI)

print(Px)
```

2.2 Once you have your dataset to work with you will need to show that you can store this data with every iteration of the data set so no data is lost.

```
import pandas as pd
import numpy as np
from datetime import datetime

DSI = np.arange(0,16)
readings = np.arange(1,33)

for Px in range(16) :
   Px = pd.DataFrame(data = np.random.uniform(size = (16,32)), columns = readings, index = DSI)
```

```
  for row in DSI:

    dt = datetime.strptime(row[0], '%Y-%m-%dT%H:%M:%S%z')

    row[0] = dt.strftime('%d/%m/%Y %H:%M:%S')

    DSI.writerow(row)


print(Px, dt.strftime('%d/%m/%Y %H:%M:%S'))
```

3 Write a function that will test the incoming data for possible strings entries

```
 import pandas as pd

import numpy as np

DSI = np.arange(0,16)

readings = np.arange(1,33)

Px = pd.DataFrame(data = np.random.uniform(size = (16,32)), columns = readings, index = DSI)

def test_Error():

  assert range(16) == 2, "Should be 2"

if test_Error == "__error__":

  test_Error()

  print("Err")
```