In [1]:

```
# Problem 1
```

In [3]:

```
import requests
url = (
    "https://raw.githubusercontent.com/changyaochen/MECE4520/"
    "master/data/random_numbers.txt"
)
response = requests.get(url)
values = [int(x.strip()) for x in response.text.split("\n") if len(x) > 0]
```

In [32]:

```
data_sheet = values       # define the values as the data sheet - data is also equal to these two
desired_sum = 5000        # define the desired sum
```

In [98]:

```
def find_pairs(data,desired_sum):  # define the find_pairs function as well as the data & desired sum
    pairs = 0                      # initialize the count

    numbers_seen = set()           # create a set that will keep track of the numbers seen during iterat

    for num in data:                     # begin iteration
        difference = desired_sum - num    # calculate the difference between the target sum and the curr
        if difference in numbers_seen:   # if the difference is a seen number, match the two
            pairs += 1                   # increment the pair count
        numbers_seen.add(num)            # add the current number to the set of seen numbers
    return pairs                         # return the number of pairs that sum to the target
```

In [99]:

```
answer = find_pairs(data_sheet, desired_sum)   # define the answer to the problem as the function
print(f"There are {answer} pairs that sum to {desired_sum} in the data sheet") # print the answer
```

```
There are 6 pairs that sum to 5000 in the data sheet
```

In [4]:

```
#Problem 2
```

In [48]:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("https://raw.githubusercontent.com/changyaochen/MECE4520/master/data/iris.csv")
```

In [49]:
```
data.head      # used class notes as a guide
data.shape
```

Out[49]:
```
(150, 6)
```

In [100]:
```
data["Species"].value_counts()  # should give you information on all the data
```

Out[100]:
```
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: Species, dtype: int64
```

In [63]:
```
data.groupby("Species")["PetalLengthCm"].mean() # average petal length - need max & min of iris-versicol
```

Out[63]:
```
Species
Iris-setosa        1.464
Iris-versicolor    4.260
Iris-virginica     5.552
Name: PetalLengthCm, dtype: float64
```

In [101]:
```
# Need to narrow the data down to the Petal Length of Iris-versicolor (Species --> iris-versicolor --> p
iv_data = data[data["Species"] == "Iris-versicolor"]  # brings us into data --> species --> iris-versico

iv_max = iv_data["PetalLengthCm"].max()    # assign a variable to the max petal length within iv_data
iv_min = iv_data["PetalLengthCm"].min()    # assign a variable to the min petal length withing iv_data

print(f"The maximum PetalLengthCm for iris-versicolor is {iv_max}") # print the answer
print(f"The minimum PetalLengthCm for iris-versicolor is {iv_min}") # print the answer
```

```
The maximum PetalLengthCm for iris-versicolor is 5.1
The minimum PetalLengthCm for iris-versicolor is 3.0
```

In [53]:
```
data.groupby("Species")["SepalWidthCm"].mean() # answer to the second part of problem 2 - Iris-versicolo
```

Out[53]:
```
Species
Iris-setosa        3.418
Iris-versicolor    2.770
Iris-virginica     2.974
Name: SepalWidthCm, dtype: float64
```

```
In [103]: average_sepal_width_cm_data = data.groupby("Species")["SepalWidthCm"].mean() # assign a variable to the
          minimum_value = average_sepal_width_cm_data[0]  # assign the minimum value variable to the data

          for value in average_sepal_width_cm_data:       # begin iteration
              if value < minimum_value:                   # look for the lowest average
                  minimum_value = value                   # assign the lowest average a variable

          print(f"Among the three species, the smallest average SepalWidthCM is {minimum_value} ")  # print answe
          print(f"Iris-versicolor has the smallest average at {minimum_value}")  # print answer
```

```
Among the three species, the smallest average SepalWidthCM is 2.77
Iris-versicolor has the smallest average at 2.77
```