

#Thasina Tabashum

Question 1

```
In [89]: myDict = {'love': 0.9, 'awesome': 0.8,
                  'cool': 0.6, 'good': 0.4,
                  'okay': 0.1, 'bad': -0.5,
                  'terrible': -0.6, 'disgusting': -0.7, 'hate': -0.9}
```

```
In [90]: comment = "I love this class! Analyzing data is so cool"
```

```
In [91]: data = re.findall(r'\w+', comment)
```

```
In [92]: print(data)

['I', 'love', 'this', 'class', 'Analyzing', 'data', 'is', 'so', 'cool']
```

```
In [93]: total = 0
         for i in data:
             if i in myDict:
                 total = total + myDict[i]
         print(total)
```

1.5

```
In [94]: print(round(total/len(data),3))
```

0.167

Calculating average happiness like above just in function

```
In [95]: # note that I also ommiting .!- this kind of expression in wordlist
```

```
In [96]: import re
         def avergave_happiness(comment):
             data = re.findall(r'\w+', comment)
             #data = comment.split(" ")
             total = 0
             for i in data:
                 if i in myDict:
                     total = total + myDict[i]
             return (round(total/len(data),3))
```

```
In [97]: comments= ["Their new shake is disgusting - I thought it was okay at first but th
                "Spiderman 2 was awesome for a sequel. I love the character developme
                "If there is one thing I hate more than the bad puns in this movie, :
                "ah, that was an okay movie, it wasn't groundbreaking, but I had a go
print(comments[0])
```

Their new shake is disgusting - I thought it was okay at first but then it left
a terrible taste in my mouth

```
In [98]: for j in comments:
        print(average_happiness(j))
```

```
-0.055
0.121
-0.1
0.031
```

Question 2

```
In [99]: import pandas as pd
import numpy as np
df = pd.read_csv('titanic_individuals.csv')
df.head()
```

Out[99]:

	Unnamed: 0	Name	PClass	Age	Sex	Survived	SexCode
0	1	Allen, Miss Elisabeth Walton	1st	29.00	female	1	1
1	2	Allison, Miss Helen Loraine	1st	2.00	female	0	1
2	3	Allison, Mr Hudson Joshua Creighton	1st	30.00	male	0	0
3	4	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	female	0	1
4	5	Allison, Master Hudson Trevor	1st	0.92	male	1	0

dropping other columns and missing values

```
In [100]: df1 = df.drop(['Name', 'Unnamed: 0'], axis=1)
df1.head()
```

Out[100]:

	PClass	Age	Sex	Survived	SexCode
0	1st	29.00	female	1	1
1	1st	2.00	female	0	1
2	1st	30.00	male	0	0
3	1st	25.00	female	0	1
4	1st	0.92	male	1	0

```
In [101]: titanic_df = df1.dropna()
titanic_df.head()
```

Out[101]:

	PClass	Age	Sex	Survived	SexCode
0	1st	29.00	female	1	1
1	1st	2.00	female	0	1
2	1st	30.00	male	0	0
3	1st	25.00	female	0	1
4	1st	0.92	male	1	0

```
In [102]: df_child = titanic_df[(titanic_df['Age'] < 18)]
df_child.head()
```

Out[102]:

	PClass	Age	Sex	Survived	SexCode
1	1st	2.00	female	0	1
4	1st	0.92	male	1	0
49	1st	14.00	female	1	1
50	1st	11.00	male	1	0
83	1st	17.00	female	1	1

```
In [103]: df_adult = titanic_df[(titanic_df['Age'] >= 18)]
df_adult.head()
```

Out[103]:

	PClass	Age	Sex	Survived	SexCode
0	1st	29.0	female	1	1
2	1st	30.0	male	0	0
3	1st	25.0	female	0	1
5	1st	47.0	male	1	0
6	1st	63.0	female	1	1

```
In [104]: grouped_survived_child = df_child.groupby('PClass')['Survived'].sum()
print(grouped_survived_child)
```

```
PClass
1st    12
2nd    25
3rd    21
Name: Survived, dtype: int64
```

```
In [105]: grouped_survived_adult = df_adult.groupby('PClass')['Survived'].sum()  
print(grouped_survived_adult)
```

```
PClass  
1st    127  
2nd     71  
3rd     57  
Name: Survived, dtype: int64
```

```
In [106]: grouped_died_child = df_child.groupby('PClass')['Survived'].count()-df_child.grou  
print(grouped_died_child)
```

```
PClass  
1st     1  
2nd     2  
3rd    35  
Name: Survived, dtype: int64
```

```
In [64]: grouped_died_adult = df_adult.groupby('PClass')['Survived'].count()-df_adult.grou  
print(grouped_died_adult)
```

```
PClass  
1st     86  
2nd    114  
3rd    205  
Name: Survived, dtype: int64
```