

In [5]:

```
from os import listdir
from pandas import read_csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# load sequence for each subject, returns a list of numpy arrays
def load_dataset(prefix=''):
    subjects = list()
    directory = prefix + 'C:\\Users\\Pujachouhan\\OneDrive\\Desktop\\Activity Recognition from Single Chest-Mounted Accelerometer\\'
    for name in listdir(directory):
        filename = directory + '/' + name
        if not filename.endswith('.csv'):
            continue
        df = read_csv(filename, header=None)
    # drop row number
    values = df.values[:, 1:]
    subjects.append(values)
    return subjects

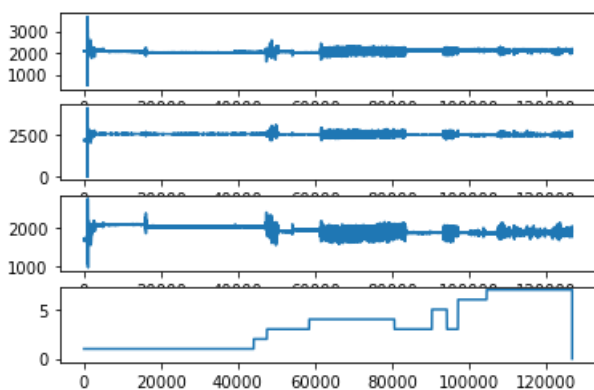
# plot the x, y, z acceleration and activities for a single subject
def plot_subject(subject):
    plt.figure()
    # create a plot for each column
    for col in range(subject.shape[1]):
        plt.subplot(subject.shape[1], 1, col+1)
        plt.plot(subject[:,col])
    plt.show()

# load
subjects = load_dataset()
print('Loaded %d subjects' % len(subjects))
```

Loaded 15 subjects

In [40]:

```
# plot activities for a single subject
plot_subject(subjects[1])
```



In [75]:

```
# returns a list of dict, where each dict has one sequence per activity
def group_by_activity(subjects, activities):
    grouped = [{a:s[s[:,-1]==a] for a in activities} for s in subjects]
    #print(grouped)
    return grouped
```

In [76]:

```
# calculate total duration in sec for each activity per subject and plot
```

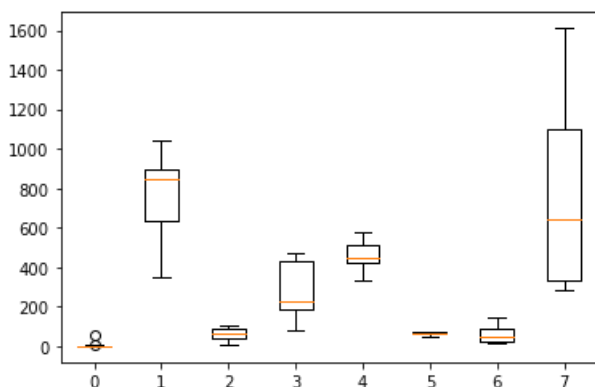
```
def plot_durations(grouped, activities):
    # calculate the lengths for each activity for each subject
    freq = 52
    durations = [[len(s[a])/freq for s in grouped] for a in activities]
    print(durations)
    #print(activities)
    plt.boxplot(durations, labels=activities)
    plt.show()
```

In [77]:

```
# load
subjects = load_dataset()
print('Loaded %d subjects' % len(subjects))
# group traces by activity for each subject
activities = [i for i in range(0,8)]
grouped = group_by_activity(subjects, activities)
# plot durations
plot_durations(grouped, activities)
```

Loaded 15 subjects

```
[[0.019230769230769232, 0.019230769230769232, 0.019230769230769232, 0.019230769230769232,
0.019230769230769232, 0.019230769230769232, 0.019230769230769232, 5.1923076923076925,
0.019230769230769232, 0.019230769230769232, 0.019230769230769232, 4.4423076923076925,
0.019230769230769232, 3.9615384615384617, 57.71153846153846], [647.6346153846154,
847.1153846153846, 1041.7307692307693, 937.5, 351.53846153846155, 1016.8269230769231,
992.3076923076923, 849.0384615384615, 801.4423076923077, 606.5384615384615, 595.7692307692307,
846.9230769230769, 629.8076923076923, 846.9230769230769, 694.0384615384615], [17.846153846153847,
66.0576923076923, 44.80769230769231, 83.46153846153847, 31.73076923076923, 9.711538461538462,
90.0, 67.11538461538461, 105.28846153846153, 92.5, 103.26923076923077, 66.53846153846153,
69.23076923076923, 67.01923076923077, 6.153846153846154], [214.98076923076923, 453.7692307692308,
168.75, 225.57692307692307, 158.07692307692307, 245.8653846153846, 152.1153846153846,
451.4423076923077, 82.5576923076923, 411.63461538461536, 258.0769230769231, 451.8269230769231,
202.28846153846155, 473.1730769230769, 217.8846153846154], [516.5384615384615, 425.9423076923077,
341.34615384615387, 579.4230769230769, 339.4230769230769, 563.75, 336.7307692307692,
426.4423076923077, 498.0769230769231, 448.0769230769231, 514.4230769230769, 423.84615384615387,
514.8076923076923, 423.75, 514.0384615384615], [61.36538461538461, 74.8076923076923,
62.01923076923077, 70.1923076923077, 65.38461538461539, 69.61538461538461, 63.46153846153846,
75.1923076923077, 51.09615384615385, 72.01923076923077, 57.69230769230769, 75.67307692307692,
56.92307692307692, 75.0, 59.90384615384615], [56.09615384615385, 143.25, 20.173076923076923,
23.076923076923077, 23.096153846153847, 28.846153846153847, 19.23076923076923, 136.53846153846155,
26.923076923076923, 36.73076923076923, 55.38461538461539, 134.40384615384616, 51.92307692307692,
115.0, 47.98076923076923], [1610.5384615384614, 427.5192307692308, 329.8269230769231,
286.5576923076923, 331.71153846153845, 298.0769230769231, 336.53846153846155, 642.9038461538462, 4
02.6923076923077, 682.5, 1492.3076923076924, 705.9807692307693, 1609.6346153846155,
649.0384615384615, 1608.8461538461538]]
```



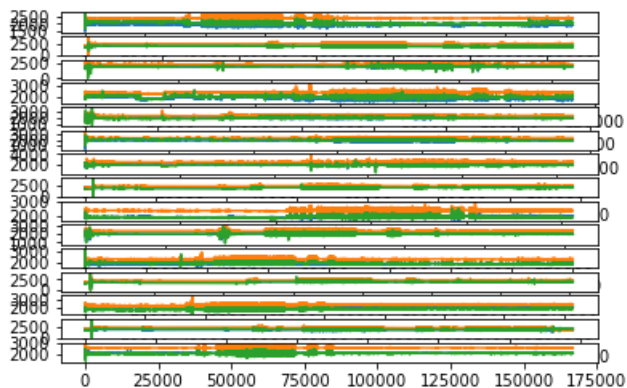
In [47]:

```
# plot the x, y, z acceleration for each subject
def plot_subjects(subjects):
    plt.figure()
    # create a plot for each subject
    for i in range(len(subjects)):
        plt.subplot(len(subjects), 1, i+1)
        # plot each of x, y and z
        for j in range(subjects[i].shape[1]-1):
            plt.plot(subjects[i][:,j])
    plt.show()
```

```
plt.show()

# load
subjects = load_dataset()
print('Loaded %d subjects' % len(subjects))
# plot trace data for each subject
plot_subjects(subjects)
```

Loaded 15 subjects

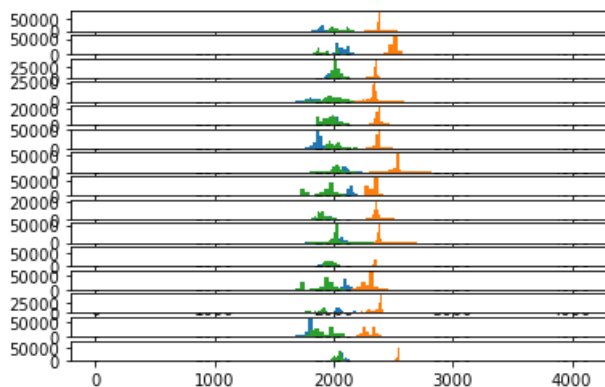


In [51]:

```
# plot the x, y, z acceleration for each subject
def plot_subjects(subjects):
    plt.figure()
    # create a plot for each subject
    xaxis = None
    for i in range(len(subjects)):
        ax = plt.subplot(len(subjects), 1, i+1, sharex=xaxis)
        if i == 0:
            xaxis = ax
        # plot a histogram of x data
        for j in range(subjects[i].shape[1]-1):
            plt.hist(subjects[i][:,j], bins=100)
    plt.show()

# load
subjects = load_dataset()
print('Loaded %d subjects' % len(subjects))
# plot trace data for each subject
plot_subjects(subjects)
```

Loaded 15 subjects



In [59]:

```
subjects[1]
```

Out[59]:

```
array([[2.082e+03, 2.165e+03, 1.702e+03, 1.000e+00],
       [2.077e+03, 2.160e+03, 1.703e+03, 1.000e+00]])
```

```
[2.077e+03, 2.100e+03, 1.703e+03, 1.000e+00],  
[2.082e+03, 2.168e+03, 1.704e+03, 1.000e+00],  
...,  
[2.105e+03, 2.467e+03, 1.908e+03, 7.000e+00],  
[2.096e+03, 2.482e+03, 1.924e+03, 7.000e+00],  
[2.102e+03, 2.495e+03, 1.939e+03, 0.000e+00]])
```

In []:

In []:

In []: