Tasks and Analysis for Simple Harmonic Motion Experiment

Before or after the experiment you will be required to add or remove some pieces of python code to a python script **(SHM1.py)** that uses the accelerometer to record the changing acceleration values. The tasks are as follows:

1. **IF else statement**
   Add a block of code inside the while loop that sets the LED screen to a specific color if each of the acceleration in the X, Y ,Z direction is greater than or equal to 0.7
   Hint: To set all the LEDs to a specific color, use **sense.set_pixels(pixel_list)** depending on the color of choice. Such as green_list or red_list.

2. **While loop**
   The while loop of this python script is an endless loop i.e. it will run continuously. Add a break statement given the condition that when the loop counter gets to 10, the program should stop looping.

3. **For loop**
   Change the while loop above to for loop. Make the loop run only 10 times

4. **Functions**

   A function called storeXYZ_values has been defined in the beginning of this python script. This function allows you to automatically save the acceleration and time interval values in a csv file. You are required to write a piece of code to call the function in a specific part of the script. The location will be indicated in the SHM1.py script.

5. **Graph your recorded values directly from the csv file**
   Create a new python script called acc_graph.py to graph your recorded x acceleration values against time directly from the csv file.

   Here's a simplified example of how matplotlib could be used to graph data loaded from a csv file.

```
#imports the necessary modules to run the script
import matplotlib
import csv
matplotlib.use("TKAgg")
import matplotlib.pyplot as plt

# arrays to store the different values of each data to be plotted
x = []
y = []
```

```
# command to read a specific csv file, take the required data needed from the file and
#append to each data to its corresponding array
with open('example.txt','r') as csvfile:
    plots = csv.reader(csvfile, delimiter=',')
    for row in plots:
        x.append(float(row[0]))
        y.append(float(row[1]))
plt.plot(x,y, label='Loaded from file!')

plt.xlabel('x')

plt.ylabel('y')

plt.title('Interesting Graph\nCheck it out')

plt.legend()

plt.show()
```