HOMEWORK 10

CS 125

due at 11:45pm (classtime) on Tuesday, September 29

Write a Python *function* to solve each of the following problems. Plan each function on paper before you implement it in code.

Prepare your solutions in a single Python file. Use comments to clearly state the problem number for each of your solutions. Provide test cases to show that your functions produce the desired output. Upload your file to the Homework 10 assignment on Moodle.

1. **Common items**: Write a function `common(list1, list2)` that accepts two lists and determines whether they have an element in common. That is, the function returns `True` if some item in `list1` is also in `list2`, and returns `False` otherwise. For example:

   `common([3, 4, 5], [6, 3])` returns `True`

   `common(["cat","dog"], [3, 4])` returns `False`

2. **Removing extreme values:** When analyzing a list of numerical data, it may be desirable to remove numbers that are far from the mean. Write a function `removeExtremeVals(values, k)` that takes a list `values` and a positive number `k` as parameters. Your function should return a new list that contains only those values that are up to `k` standard deviations away from the mean. You may use your `meanSD` function from class to compute the mean and standard deviation.

   Specifically, if the mean is $\bar{x}$ and the standard deviation is s, then your function should return only those values x that satisfy $\bar{x} - ks \leq x \leq \bar{x} + ks$.

3. **Sieve of Eratosthenes:** The Sieve of Eratosthenes is an algorithm for making a list of all prime numbers between 2 and some limit. The algorithm is given on the next page.

   Write a function `sieveErat(limit)` that returns a list of all prime numbers less than or equal to `limit`.

Here is the sieve of Eratosthenes algorithm in pseudocode:

Start with the list [0, 1, 2, 3, …, limit].
Replace the 1 in the list with 0.
Set *k* equal to 2.
While *k* is less than the limit:
    Find all multiples of *k* (larger than *k*) in the list and set them to 0.
    Set *k* equal to the next nonzero number in the list.
Remove all zeros from the list.
The list now contains only prime numbers.

4. **Anagrams**: An anagram is a word or phrase made by transposing the letters of another word or phrase. For example, "parliament" is an anagram of "partial men" and "software" is an anagram of "swear oft". Write a function that accepts two strings and returns `True` or `False`, depending on whether the strings are anagrams or not. Ignore spaces and punctuation. For example:

    `anagrams("parliament","partial men")` returns `True`

    `anagrams("cat","dog")` returns `False`

    As part of your solution, you may wish to call your `letterFrequencies` function from class.