CS:1210 Discussion Section Examples

In the discussion sections this week (Feb 24-28) some subset of the following problems were discussed. Please see the course website for solutions to these problems.

1. Write a function called `concatenate` that takes a list of strings as a parameter and returns a long string that is the concatenation of all the strings in the list, taken in order. For example, if the given list is `"These", "are", "hello"` then the function would return `"Thesearehello"`.

2. Write a function called `isSorted` that takes a list of numbers as a parameter and returns `True` if the list of numbers is sorted in ascending order and `False` otherwise. For example, if the given list is `[3, 8.5, 8.5, 11, 22]` then the function would return `True`; if the given list is `[3, 8.5, -11, 22]` then the function would return `False`.

3. Write a function called `subsetOf` that takes two lists and returns `True` if every element of the first list is also in the second list; otherwise the function returns `False`. For example, if the first list is `[3, 8.5, -22]` and the second list is `"hello", -22, "hi", 8.5, "goblin", 3` then function would return `True`. On the other hand, if the first list is `[3, 8.5, -22]` and the second list if `"hello", -22, "hi", "goblin", 3` then the function would return `False`.

4. Define a function called `gradeDistribution` that takes a list of exam scores and returns a list that contains the distribution of these scores. To be more precise let us suppose that the exam scores are out of 100 and therefore these are floating point numbers in the range 0 through 100 (inclusive of 0 and 100). The distribution of the scores we want you to compute is the number of scores that are in each of the ranges `[0, 10]`, `(10, 20]`, `(20, 30]`, `(30, 40]`, `(40, 50]`, `(50, 60]`, `(60, 70]`, `(70, 80]`, `(80, 90]`, and `(90, 100]` (we are using `(A, B]` to denote the range $A < s \leq B$ and $[A, B]$ to denote $A \leq s \leq B$). In other words, the first element in the list returned by your function should be the number of scores in the range `[0, 10]`, the second element should be the number of scores in the range `(10, 20]`, etc. You should use the following function header:

```python
def gradeDistribution(examScores):
```