David Tsui
CS 1671 Assignment 1

1. **For each of the specified types, what are some variations that your program can cover?**
   The program can cover the basic requirements of handling normal numbers, dollars, dates, fractions, and percent.

   As well, it handles the following augmentations:
   - Ordinals
   - dollars in different formats such as $ 1 million
   - dollars in combination with decimals
   - dollars with cents that are greater than 2 digits
   - percent with fractions
   - percent with decimals
   - dates with just month and day
   - dates with just month and year
   - dates with month, day, year
   - dates with month concatenated
   - dates with prepositions 'in' and 'until'
   - numbers with comma
   - numbers with decimal

2. **What was your strategy for identifying variations, given that you probably don't want to read every sentence?**
   My strategy is to use regular expression to identify which category of numbers it belongs to. Each category will have several different regular expressions to handle the different variations within each category of numbers. Python has a findall regex which I use to substitute all the matched regex combination at once before moving to the next regex combination.

   To handle the different ambiguous cases, I used a pipeline/cascading regular expression translation to first filter out the more specific cases followed by the more general cases. This filtering is applied both within each category and outside the category as well.

   For example, within the category of percent, I first look for the more specific, decimal numbers with percent then look for normal numbers with percent and finally look for any percent and translate each before looking for the next regex combination.

   Overall, I look for all the special cases first before finally looking for the normal numbers case.

3. **Were there difficult cases that your program cannot correctly handle using regular expression matching? If so, what additional tools do you think might be needed?**
   Yes, my program can only determine what the category is by looking at regex. This means that inference through context or meaning will not be possible.

   For example, in the sentence:
   "He was born in 1994. 1994 saw a lot of changes in the global economy."

The program can identify that 1994 in the 1st sentence is a year because of the word “in” but it will not recognize the following 1994 is also a year. To correctly parse a sentence like this, I will need additional tools of finding relationships between sentences and identifying sentence structures.