CIS195 Homework 01.a

Calculator Prep

Due: ***Monday, January 28th at 11:59PM***

Goal: To start planning ahead for your first iOS homework assignment.
What you will submit: The link to your repository from Homework 00, with the updated README.

If you haven't guessed from Homework 00, Homework 01 is to build your very own calculator app. Although we haven't started learning about the Objective-C programming language yet, there is nothing stopping us from planning out how you will implement the calculator logic (i.e. the computation).

The amount of detail with which you decide to lay out your plan is entirely up to you, but the more detailed you make it, the easier your life will be when you start translating your plan to Objective-C for next week's homework assignment.

Looking ahead, below are the basic specifications for the calculator that you will be building.

Basic calculator specs:
- Supports *infix* notation (see **git.to/infix**).
- Supports addition, subtraction, multiplication, and division.
- Supports decimal numbers.
- Supports negative numbers.
- Abides by correct order of operations.
- Handles divide by zero elegantly (e.g. ∞, NaN).
- Allows user to clear their input.

In general, we give generous amounts of extra credit for optional features and any features that you think would be useful and decide to implement (we like creativity!).

Optional (Extra Credit)
- Supports parentheses for changing order of operations.
- Supports exponentiation.
- Supports input like **π**, **e**, etc.
- ... and whatever else you think would be useful!

The **input** that you will be handling is a *string* representing user input, such as the examples below.

"1 + 2"
"2 x 3 + 1"
"1 + 15 ÷ 4 – 3"
etc.

Your **output** should be the resulting number that corresponds to the computation expressed by the input string.

3
7
1.75
etc.

1. In your README from Homework 00, describe how you will compute the output from the input. Don't worry about what the exact names of the data structures that you will use. Objective-C has all the basic data structures such as arrays, dictionaries, utility functions such as string parsers, and classes with private and public variables and methods. You may use pseudo-code, words, or even a different programming language that you are comfortable with. You can use one class, two classes, multiple functions, or even just a single function that does everything (for now, at least). Please try to use your good judgement about proper object-oriented design.

2. Once you are done editing your README, you can push your updates to the repository by running the following commands:

```
git commit -am "added calculator plan to readme"
git push origin master
```

3. Submit your repo link at **git.to/hw01a**.

4. You're done!