CIS 212: Project #2C
Assigned: October 16, 2019
Due: October 26, 2019
(which means submitted by 6am on October 27, 2019)
Worth 6% of your grade

Please read this entire prompt.

Assignment:
1) Write a C function that parses a string that represents a floating point number (example: “3.68”) and turns it into a real number (i.e., 3.68). (If confused by this, see the starter code.)
2) Quite a few hints are below.

This project will be graded by running the Unix script “proj2C_checker” on the VirtualBox. proj2C_checker is available on the course website.

If your program fail the checker script on the VirtualBox, then you should expect less than half credit.

There are various corner cases in implementing this. For example, parsing 3.23487293489 will break most implementations since “23487293489” is bigger than 2^31. I am not expecting you to figure out all of these corner cases. If you pass “proj2C_checker,” then you will get full credit, provided you actually have implemented parsing (explicitly: no atof, scanf, etc.).

If the proj2C_checker program shows any difference, you will get less than half credit.

What should you upload?: Just a single file, which is your C source code.

Big hints:
- As always, compile frequently and test your code often.
- DO NOT try to implement everything all at one time. You will likely end up with a mess that will be difficult to debug.
- Break this into small parts. Start with a hello world. Then get the value of the first digit. Then positive integers. Then positive or negative. Then add fractions. Then run the checker script. You should only run the checker script after you feel your program is “ready.”

Other hints:
- The negative sign is only permitted as the first character. If you check for it first, then the rest of your code doesn't have to deal with it.
- Make sure to use “double” (8 bytes) rather than float (4 bytes)
- When I found a “.”, I called a new function on the remainder of the string. I called my new function “CalculateFraction()”. 
- When looking at a single character, you can calculate its value with a trick:
  - The trick is c-‘0’ where c is a character. Note that is a quoted ‘0’, which is “48” in ASCII. ‘1’ is 49, ‘2’ is 50, etc. So if c is ‘5’, then ‘5’ is 53, and 53-48 is ... 5.
- If your program never stops then your logic isn’t right. Add print statements inside your while loops or for loops.

This code:
```c
int A = 5;
int B = 10;
double fraction = A/B;
```
will produce a value for fraction of 0!

Instead, do:
```c
double fraction = (double)A/(double)B;
```
(this is called casting)