TinyCalc - Basic constructs, functions and pointers in C
Due 11:59pm Monday February 8th, 2016

For this assignment you will implement TinyCalc - a simple terminal-based calculator application. The calculator can add, subtract, multiply, divide and compute x to the power y where y >= 0. TinyCalc also maintains the results of the last five computations. Each of these results can be accessed and used in subsequent operations. Computations are performed by typing an operator followed by the operand. The operation is performed on the operand and the most recent result. The most recent results are all initialized to zero at start up. You can find an example executable file at
`~mharmon/cs277/bin/tinycalc`

See the following sequence of commands as an example of how the application works:

Addition:
```
>+1
1.00
```

Subtraction:
```
>-6
-5.00
```

Multiplication
```
>*4
-20.00
```

Division
```
>/2.5
-8.00
```

Power
```
>^4
4096.00
```

Memory Operations
TinyCalc maintains the five most recent results which can be accessed by typing `m` followed by a numbered memory location:

0: previous computation
1: 2nd previous computation
2: 3rd previous computation
3: 4th previous computation
4: 5th previous computation

Example: display and load the 4th previous computation:
```
>m3
-5.00
```

Example: use the 5th previous computation:

```
>m4
1.00
>*99
99.00
```

Example: memory locations outside of the range, default back to the previous computation:

```
>m657
99.00
```

Assignment Files

There are three files with the assignment:

`tinycalc.h` this is the header file that defines the core features of the application and the key constants, data structures, etc. open it to read about the behavior of each function but **DO NOT MODIFY** it.

`tinycalc.c` this source file is where you implement the five functions defined in `tinycalc.h`. Blank implementations of each have been provided to get you started.

`driver.c` this source file is where you should put your main application code. The welcome message and initial prompt have been provided for you.

Compiling and Running your Solution

To compile, type the following from the command line:

```
gcc –std=c99 -o tinycalc *.c
```

to run type:

```
./tinycalc
```

Requirements and Notes

1. You may not use `math.h` for this assignment.
2. Global variables are not allowed. Use functions to pass data between scopes.
3. For full credit, the output of your application should match the output of the sample executable **EXACTLY.**
4. For grading, I will be unit testing each function defined in `tinycalc.h`
5. You can assume that for grading, I will pass a character followed by an integer or double value as input. I may, however, pass invalid commands or nonsensical numbers so be sure your code can handle things gracefully. If in doubt, see what the sample executable does! A couple example of valid bad input:

   ```
   c8
   m-2.5
   ```