Project 01 | Command Line Calculator 1.2-1.5

This project is intended to practice working with variables, reading input, storing & reusing data, functions, and understanding control flow. This is an extension of project 1, the command line calculator, and includes four additional features as added levels of complexity.

Spec 1.2-1.5

1. *Basic calculator operations; see 1.1 spec document. Should be completed prior to beginning this document's tasks.*
2. Change the user prompt to simply be "Enter Instruction:", but add a help command. If the user types "help" ":help" or "h", print out the following in a nice format:

```
A command line calculator. Computes basic mathematical operations of two operands:
    +       Addition
    -       Subtraction
    *       Multiplication
    /       Division
    :exit   Exits the calculator
    :help   Prints help information
```

3. Allow the user to use the previously computed result. When entering an operand, the user may type "ANS" to indicate they would like to use the result of the previous calculation. If this is the first operation being computed, print an error message ("No prior answer to use for computation") and re-prompt the user for input. One or both operands may be "ANS".
4. Add three additional operators, *and add them to the help menu described in (2)*:
    - `^`: Power operation, first operand is the base and second operand is the exponent. Use `Math.pow` to compute.
    - `%`: Modulo operation, first operand is the dividend and the second operation is the quotient. Returns the remainder of the division performed.
    - `|`: Absolute value, takes only one operand and returns its absolute value.
5. Add the ability for the user to save a value to memory:
    - Add new operation `:MEM_SAVE` which prompts the user for one operand, a number, and saves that result into memory. User should be able to enter "ANS" to retrieve the prior computation's result per (3).
    - Add new operation `:MEM_LOAD` which retrieves and prints the saved memory for the user's reference.
    - Add new valid operand "MEM" when entering operand information. Retrieves the user's saved memory value and uses it for the computation.

Sample Output

- Left as an exercise to the reader.

Submission

- Name your Python program `01-5_calculator_<lastname>_<first-initial>.py`;
  e.g. `01-5_calculator_campbell_z.py`
- Compress the file into a zip of the same name, either from the finder or the command line.
- Email it to `zacharyhcampbell@gmail.com`.