You’re going to implement a calculator with a “paper tape” that exports to a file.

# The Calculator

Your calculator will work as follows:

1. Prompt for, and accept, a floating-point number.
   * Allow the user to enter a number, or:
     + The word “pi”
     + The letter “e”
     + The letter “c”
     + The word “ca”
     + The letter “q”
     + The letter “b”

The result of “pi” and “e” should be obvious; for the others, see below.

1. Prompt for, and accept, an operation. The operations are as follows:

|  |  |  |
| --- | --- | --- |
| + | addition | binary |
| - | subtraction | binary |
| x | multiplication | binary |
| / | division | binary |
| ^ | exponentiation | binary |
| sin | sine | unary |
| cos | cosine | unary |
| tan | tangent | unary |
| arcsin | inverse sine | unary |
| arccos | inverse cosine | unary |
| arctan | inverse tangent | unary |
| root | square root | unary |
| abs | remove sign | unary |
| inv |  | unary |
| log |  | unary |
| log2 |  | unary |
| sto | store (only accepts 0-9) | binary |
| rcl | recall (only accepts 0-9) | unary |
| c | clear | special |
| ca | clear all | special |
| q | quit | special |

1. If the operation is a binary operation, prompt for, and accept, another number.
2. Print the result, to the screen.
3. Return to step 2.

# STO and RCL

Your calculator should provide ten storage variables to the user. They work as follows:

The STO operation stores a result in a variable. With a number displayed, typing STO prompts for a number 0-9.

* If it gets a number 0-9, it stores the previous number in that variable.
* If it gets anything else, it displays an error.

The RCL operation recalls a result in memory. With a number 0-9 displayed, it attempts to recall from that storage variable.

* If a number 0-9 is displayed, RCL recalls the number from that variable and displays it.
* If anything else is displayed, it displays an error.

Start all the storage variables at zero. Whenever you read “ca”, reset them all to zero.

# Notes

* Align all outputs. This means you’ll need to pick a maximum number of digits both before and after the decimal. Can you figure out what that should be?
* For any mathematical or other error, print “error” and return to step 1.
* Any time you read “c”, print “0” and return to step 1.
* Any time you read “ca”, clear storage variables 0-9 and return to step 1.
* Any time you read “q”, close the output file and exit.
* Any time you read “b” for a number:
  + Print the previous entered number or result, and prompt the user to type “b” again or just hit enter.
  + If the user hits enter, use that result for that number; if the user types “b” again, repeat the above, *including* prompting them to type “b” or hit enter again.
  + Note that this means you need to store all previous numbers and results in memory.

# The Tape File

* Output every entered number, the code and name of every operation, and every result.