Reverse Polish Notation Calculator

Example of RPN:

```
1 2 * 3 4 * *
```

This is the equivalent in regular notation:

```
(1 * 2) * (3 * 4)
```

Numbers are ingested, converted into a double representation and then pushed onto a stack. This stack is implemented using a regular C array that is where the farthest most element is the top of the stack.

Operations

Math

The following mathematical functions are supported:

- SIN
- COS
- TAN
- ASIN (arcsin)
- ACOS (arccos)
- ATAN (arctan)
- POW
- EXP
- ADD ('+')
- SUB ('-')
- MUL ('-')
- DIV ('-')
- MOD ('%')
- FLOOR
- SQRT

These operands are considered keywords, therefore they cannot be used for variable assignment. `ADD` through `MOD` are called using the character adjacent to the operand code name. If a number is pushed onto the stack and one of these operands are misspelled, then the program will throw a `GARBAGE` operand code, which will then signal an error message and subsequent clearing of the stack.

Stack

The following stack operations are supported:

- POP
  - Take the top most element off the stack
- PEEK
  - See the top most element of the stack without modifying the stack pointer
- SWAP
  - Swap the top two most elements with each other
- CLONE
  - Make a copy of the stack

Variable Assignment and Retrieval

The way we set a variable can be seen in the following example:

```
10 =variable
```

Please note that the value of the variable needs to precede the `=variable` assignment. The value is pushed onto the stack as it can be used for other operations. If nothing else comes after the variable assignment, the stack will be cleared.

We can retrieve variables with the symbol `?` as seen here:

```
?variable
```

Where the value of variable is retrieved from a hash table containing a key (variable name) and a subsequent value (value assigned above)

Both of these can be chained together in one line like so:

```
10 =variable
10 =variable_two
?variable ?variable_two * =variable_three
?varible_three
PEEK
```

Here we assign both `variable` and `variable_two` to contain 10, then we multiply the result and store that value into `variable_three`. We can now see the value of of `variable_three` by calling `PEEK` without altering the stack.

Constants

The only constant currently supported is pi which can be called by `PI` and will be pushed to the stack