General Information

For this lab, you will write a simple pocket calculator program. The program will be able to display the current value on the screen of the calculator. You will store the current value of the calculator in a variable. The initial value of the calculator (i.e., the initial current value) is 0.

Do not write out the entire lab assignment and only then try to debug it: this almost never works. If you’re new to programming, you shouldn’t, as a rule, write more than five lines of code between trying the new code out to see if it does something sensible.

You will get credit for the lab if you make reasonable progress toward completing it. Credit may be given for programs that accomplish only some of the tasks assigned. The TAs are here to help you. If you are stuck, ask for help!

The Tasks

1. In the if __name__ == 'main' block, write code that displays the following:
   Welcome to the calculator program.
   Current value: 0

2. Write a function whose signature is
   display_current_value()
   , and which displays the current value of the calculator. Test this function by calling it to make sure it works.

3. Write a function whose signature is
   add(to_add)
   , and which adds to_add to the current value. Test the function add by calling it and display_current_value. **Hint:** when modifying global variables from within functions, remember to declare them as global.

4. Write a function whose signature is
   multiply(to_mult)
   , and which multiplies the current value by to_mult. Test the function.

5. Write a function whose signature is
   divide(to_div)
   , and which divides the current value by to_div. Test the function. What values of to_div might cause problems? Try them to see what happens.

6. Pocket calculators usually have a memory and a recall button. The memory button saves the current value and the recall button restores the saved value. Implement this functionality.

7. Implement a function that simulates the Undo button: the function restores the previous value that appeared on the screen before the current one.
If you have time

**Without using the ** operator**, write a function whose signature is

\[ \text{pow}(\text{exponent}) \]

, and which raises the current value to the power of \( \text{exponent} \).

Hint: recall how we used a loop to draw a series of n-gons:

```python
#repeat the block below 5 times:
for i in range(5):
    draw_ngon(n, 20)
    turtle.penup()
    turtle.forward(40)
    turtle.pendown()
    #increase n by 1
    n = n + 1
```

*If you're already comfortable with loops*, try to make your function as efficient as possible.