LC-3 Calculator:

Basic steps:

1) Get Input
2) Convert ASCII numbers to integers
3) Determine innermost function (when dealing with parentheses)
4) Determine operations
5) Apply operations
6) Display output

Example equation

((5 + 7) * (9 – 6)) / 1 8

Steps to solve

((5 + 7) * (9 – 6)) / 1 8
= (12 * (9 – 6)) / 1 8
= (12 * 3) / 1 8
= 36 / 18
= 2

How the program does this?

- Uses three arrays ~ Input, number, and operation

Input Array holds the string input by the user, in ASCII form.

Example (25+7):
x28 ~ (
x32 ~ 2
x35 ~ 5
x2B ~ +
x37 ~ 7
x28 ~)

The ASCII numbers are then converted into integers, and stored in the number array. The input array is then updated. All the previous ASCII numbers are replaced with pointers to their storage location in the number array

Example (25+7):

Input Array:	Number Array:
x28 ~ (	#25 ~ At location x4000
x4000 ~ #25	#7 ~ At location x4001
x20 ~ space	
x2B ~ +	
x4001 ~ #7	
x28 ~)	

Then the inner most parenthesis is found, by using a stack counter. The equation inside of these parentheses is treated as its own equation for now, as if it was the only thing entered.

Then the operations are found and added to the operations array. Elements are of the form

Term 1
Operation
Term 2
Output

Example (25+7):

Input Array:	Number Array:	Operation Array:
x28 ~ (	#25 ~ At location x4000	x4000 ~ at x6000
x20 ~ space	#7 ~ At location x4001	#10 ~ at x6001
x20 ~ space		x4001 ~ at x6002
x6003 ~ Output of operation		#0 ~ Holding spot for answer (operations haven't been performed yet) at x6003
x20 ~ space		
x28 ~)		

Then the parentheses are replaced with spaces, and then the process starts over again. Continues until there are no operators left. Then the operations take place

Once the operations are performed, the output is then displayed