

## Exercise 01: Calculator (`calc.c`)

Skills trained: Loops, conditions and calling standard functions.

### Calculator

Implement a calculator. The calculator should be able to execute seven instructions in total and should use decimal numbers (`double`).

#### Details

First read a decimal number, then read an arbitrary number of instructions (all separated by space or newline). These instructions are executed and the result is used for the next instruction. Finally, after = has been printed, your program should print `Result:`, followed by the final result and terminate.

Your program should handle the following instructions: `+` (addition), `-` (subtraction), `*` (multiplication), `/` (division), `neg` (negate) and `sqrt` (square root). The instructions `+` (addition), `-` (subtraction), `*` (multiplication) and `/` are followed by another number, the second argument.

The result should be printed with three decimal numbers.

#### Hints

- Find online some documentation on how to execute the function `sqrt` (look for `math.h`).
- Repeat last semester's chapters on string comparison.
- There is no necessity for error handling on faulty input.
- `scanf` is sufficient for input (to read a double use `%lf` for "long float").
- Look up the format of the `printf`-format string to get the correct amount of decimal numbers.
- If you compile your program on the command line, you might have to link the math library. For GCC, this is done with the `-lm` argument. The full command using gcc thus is `gcc calc.c -o calc -lm`.

#### Example:

##### Input:

- The number 2.1 is negated.

2.1 neg =  
Result: -2.100

- First, 1.1 is added to 2.1, then the result is negated.

2.1 + 1.1 neg =  
Result: -3.200

- 1 is divided by 9. The numbers are always read as decimal numbers.

1 / 9 =  
Result: 0.111

- First, 3 is added to 2, then the result, 5, is multiplied by 4, then, 2 is subtracted which gives 18, and the result is finally divided by 3.

2 + 3 \* 4 - 2 / 3 =  
Result: 6.000

- Square root of 5 \* 5

5 \* 5 sqrt =  
Result: 5.000