

Arithmetic Operations

Let's get started with some simple numerical computations in Bash.

As can be observed from the examples below, there are several ways of making simple numerical calculations in Bash. Just trying to echo an expression wrapped in quotation marks will not work. Wrapping the expression in double parenthesis `$((..))` evaluates it, but this is confined to integer computations. To evaluate expressions involving decimal places (floating points) "bc -l" is very useful.

```
~$ echo "5+5"
5+5
~$ echo "5+5" | bc
10
~$ echo "5+5" | bc -l
10
~$ echo "5+5.2" | bc -l
10.2
~$ echo "5+5.2" | bc
10.2
~$ echo "3/4" | bc
0
~$ echo "3/4" | bc -l
.75000000000000000000
~`$ echo `$((3+3))
6
```

To display the final result by rounding it to a certain number of decimal places, "printf" with a format specified can accomplish the task by specifying the "scale" (number of decimal points). Note that the ordering of the numbers matters in this case, as demonstrated below.

```
~$ echo "scale = 2; 10 * 100 / 30" | bc
33.33
~$ echo "scale = 2; 10 / 30 * 100" | bc
33.00
~$ echo "scale = 2; (10 / 30) * 100" | bc
33.00
```

'Expr' is another way to accomplish such tasks.

```
~`$ echo `$ (expr 5 + 5)
10
~`$ echo `$ (expr 5 - 5 + 2 )
2
~`$ echo `$ (expr 5 - 5 + 2 / 3 )
0
~`$ echo `$ (expr 5 - 5 + 2 / 1 )
2
```

Be careful with spacing in such expressions! Bash is very sensitive to them.

Task

We provide you with expressions containing `+`, `-`, `*`, `^`, `/` and parenthesis. None of the numbers in the expression involved will exceed **999**.

Your task is to evaluate the expression and display the output correct to **3** decimal places.

Sample Input 1

```
5+50*3/20 + (19*2)/7
```

Sample Output 1

```
17.929
```

Sample Input 2

```
-105+50*3/20 + (19^2)/7
```

Sample Output 2

```
-45.929
```

Sample Input 3

```
(-105.5*7+50*3)/20 + (19^2)/7
```

Sample Output 3

```
22.146
```

Recommended Resources

A quick but useful tutorial for Bash newcomers is [here](#).

Handling input is documented and explained quite well on [this page](#).

A detailed coverage of arithmetic and bitwise operators is available [here](#). Multiple approaches to math in shell scripts are described [here](#).