Excercise 1 [Simple calculator] Write a very simple computer algebra program which processes commands reading line by line using fgets (see the manual page for details). This simple calculator is expected to process only positive integers and only operations with one or two operands are allowed, e.g. \(2+3\) or \(5\) are allowed, but \(1+2+3\) is not. The operators are expected to be binary always (for instance, cases like \(+15\) are not allowed). In principle, you are requested to allow only one operator: +, with obvious meaning.

Your program should be able to handle the following inputs:

- **exit** or **quit** should quit the program;
- **restart** or **reset** should initialize all the variables to 0;
- a **positive integer** number, e.g. 5, should produce the expected output, i.e. printing out 5 to the screen;
- the operations with **positive integers** should also print out the result (e.g. given \(1+3\) as input, the output should be 4);
- allowed variable names are a, b, ..., z (hint: use an array with 26 elements to store the values of the variables);
- a must print out the value associated with variable a;
- combinations like a+1, 3+b or a+b should produce the corresponding output to the screen;
- a=17 ought to assign to variable a the value 17;
- a=b must assign to variable a the contents of variable b;
- a=7+1, a=3+b or a=a+b should assign to the variable a the result of the right hand side of the = symbol.

In case of a syntax error your program should complain, but be able to continue.

Two advices:

- It may be a good idea to remove all the spaces before analyzing the input.

**Important:** think carefully about the strategy/algorithm that you want to implement before you start coding. If you don’t analyze the problem in advance, it will be much more difficult to succeed.

**Optional (⋆):** Extend your program such that the multiplication of two positive integer numbers is allowed.

**Optional (⋆⋆):** Modify your code to allow operations with non-integer positive numbers like \(1.23\). Include the division operator /.

**Optional (⋆ ⋆ ⋆):** Add negative numbers to the game.