# TP5 - KEY VALUE STORE
# (2)

"parsing text input"

## Parsing text input

In this TP you will prepare the text input functionality of your Store. String literal commands will be sent at a later implementation through sockets, and based on their meaning the respective functionality of TP4-01 will be executed.

in detail:

1. Check **notes.org** (provided) to see the syntax for communicating with the Store. Your job is to execute the previous functionality based on user text commands.
2. Arbitrary user input (null terminated strings) have to be properly parsed byte-by-byte and decomposed for processing based on the given syntax.
3. Store replies have to be returned in strings (VAL, UND, ERROR, DON)
4. Implement your string parser and alter the previous main function used for testing, to support execution (or not if there are syntax errors from "users") of commands using the given syntax.
5. Make sure to check **string.h** methods for string comparisons, string tokenizing and general string manipulation.
6. To make your life easier decompose the user string literal into its logical parts (check the empty struct in command_parser.h)
7. Alter your Makefile to include command_parser.h into the compilation chain.
8. Run the command of the user (check new version of store.h)
9. Check https://chamilo.unige.ch/ for the new project files and stubs.

user input -> parse -> decompose -> run_command -> store functionality

**GOOD LUCK!**

Heed this advice. "Do not google C strings. Instead google ANSI C strings."