

# Project 2 - Logic

---

## Read input data

- Using `read_input` method to read data from input file
- Split clauses that separate by "OR"
- The method return:
  - `alpha` for negating and manipulating
  - list `clauses` for sorting

## Write output result

- Using `write_out` method to write the result and the conclusion into a file
- With the empty result, that means there is no generated clauses during the manipulation or the alpha clause that negate with one of clauses in knowledge base at the beginning so we write the conclusion only
- With the non-empty result we join items of a clause with "OR" and write into the file. After all of the result written, we write the conclusion into the file

## Resolution Method

- Using `pl_resolution` method to manipulating and solve the question with the input data.
- First, we negate the `alpha` and if it has "--" then we remove them until the `neg_alpha` has "-" or none. Then append the `neg_alpha` into KB. (Note that: we should do the same with each clause in the KB, but the submitted code did not implement it.)
- Second, we remove the duplicate clauses in the KB and sort each item of each clause, and then sort clauses in the KB.
- Third, with the loop, we go through the clause in the KB and resolve it with the rest of sentence from the index of the considered clause (by this, we do not need to resolve the clause paired second time except the last one).
  - Because we do not need the duplicate clause in the `new_clauses`, we use `set()` to store them, and because a set is not hashable so we using `frozenset()` to implement some manipulations that require hashable items.
  - if `new_clauses` is not empty, then we remove the opposite literals in each clause, remove empty clauses, and remove clauses that exist in the KB already
- Finally, we update the KB and the result

## Test cases

- TC\_01: `input1.txt` with the single literal clause and multi-literals clauses to check whether the solution work fine with single clauses
- TC\_02: `input2.txt` with multi-literals clauses to check whether the solution work fine with complex clauses
- TC\_03: `input3.txt` with the complex clause and increasing the quantity of clause
- TC\_04: `input4.txt` with complex clause in both KB and alpha to check the solution work fine with the complex alpha

- TC\_05: `input5.txt` with the complex clauses and the alpha is nagate with one clause in KB to check the `write_output` method work fine with empty result