Prob. 1	Prob. 2

Team members: Küng, Pirelli, Schubert, Dousse, Vu

Problem 1.

Let us prove that *TWICE_SAT* is NP-complete by reducing 3SAT to it. Given a 3CNF formula ϕ with k variables, we create the formula $\phi' := \phi^{(}x_{(k+1)}\ v\ not\ (x_{(k+1)})$. Clearly, if ϕ has at least one satisfying assignment, ϕ' must have at least two satisfying assignment, one where $x_{(k+1)}$ is true and one where it is false. Therefore, *TWICE_SAT* can be reduced to 3SAT, which means *TWICE_SAT* is NP-complete.

Problem 2.

Let us define our algorithm this way :

```
loop for each variable $x$:
    remove all clauses containing variables that only appear once

    remove all clauses containing literals that appear twice (in the entire formula)

    remove all clauses containing variables that appear twice in the clause

    find clauses $c_1$ and $c_2$ such that $c_1 = X\ OR\ Yandc_2 = \ not(x)\ OR\ Z$ for some $Y, Z$ (clauses)

    if $c_1$ and $c_2$ exist, remove $c_1$ and $c_2$, add $c_3 := \ Y\ OR\ Z$

    if any clause is empty and the loop isn't finished, the formula is not satisfiable.

    if the loop finishes, the formula is satisfiable.
```