Équipe CC

1. Créer un dépôt git et m'inviter dessus.
2. Lire et comprendre l'algorithme suivant :

Algorithm 4.1.1: Congruence-Closure

Input: A conjunction φ^{UF} of equality predicates over variables and uninterpreted functions

Output: "Satisfiable" if φ^{UF} is satisfiable, and "Unsatisfiable" otherwise

1. Build congruence-closed equivalence classes.
   (a) Initially, put two terms t_1, t_2 (either variables or uninterpreted-function instances) in their own equivalence class if $(t_1 = t_2)$ is a predicate in φ^{UF}. All other variables form singleton equivalence classes.
   (b) Given two equivalence classes with a shared term, merge them. Repeat until there are no more classes to be merged.
   (c) Compute the *congruence closure*: given two terms t_i, t_j that are in the same class and that $F(t_i)$ and $F(t_j)$ are terms in φ^{UF} for some uninterpreted function F, merge the classes of $F(t_i)$ and $F(t_j)$. Repeat until there are no more such instances.
2. If there exists a disequality $t_i \neq t_j$ in φ^{UF} such that t_i and t_j are in the same equivalence class, return "Unsatisfiable". Otherwise return "Satisfiable".

3. Itérer l'algorithme sur l'exemple suivant :
   $x_1 = x_2 \wedge x_2 = x_3 \wedge x_4 = x_5 \wedge \neg(f(x_1) = f(x_3))$
4. Implémenter l'algorithme présenté à la question 2.