INPUT: Formula F of the form $G_1 \wedge \ldots \wedge G_n$, where each G_i can be of the form:

- $t_1 \ \diamond \ t_2$. Possible forms of t_1, t_2 and $\diamond$ are presented in the table:

t_1	t_2	$\diamond$
arithmetic term[a]	arithmetic term	$\{<, <, \leq, \geq, =, \neq\}$
record	*record*	$\{=, \neq\}$
record	*symbolic constant*	$\{\neq\}$
record	*arithmetic term*	$\{\neq\}$
variable	*record*	$\{=, \neq\}$
variable	*symbolic constant*	$\{<, <, \leq, \geq, =, \neq\}$
variable	*arithmetic term*	$\{<, <, \leq, \geq, =, \neq\}$

[a]for definition of arithmetic term, see algorithms.pdf

- $t \in D$. Possible forms of t and D are presented in the table :

t	D
arithmetic term	$n_1..n_2$, $\{n_1, \ldots, n_k\}$, where $n_1, \ldots, n_k$ are numbers
non-arithmetic variable[a]	$\{t_1, \ldots, t_n\}$, where all t_i are arbitrary ground terms.

[a]for definition of arithmetic variable, see algorithms.pdf

- $\neg(t \in D)$ the possible forms of t and D are the same as in $(t \in D)$.

OUTPUT: True if F is satisfiable and false otherwise.
BEGIN. $BODY :=$ **true**; $\Pi_{prolog} := \emptyset$
For each G_i in $G_1 \wedge \ldots \wedge G_n$

1. G_i **of the form** $t_1 \diamond t_2$ **or** $\neg(t_1 \diamond t_2)$

   $BODY := BODY \wedge G_i$

2. G_i **of the form** $(t \in D)$

   (a) **t is an arithmetic term, and D is of the form n1..n2**
   $BODY := BODY \wedge t \ in \ n1..n2$

   (b) **t is an arbitrary term, D is of the form** $\{t1, t2, \ldots, tn\}$
   $\Pi_{prolog} := \Pi_{prolog} \cup set\_d(X) : -member(X, [t1, t2, \ldots, tn]).$[1]
   $BODY := BODY \wedge set\_d(t)$

3. G_i **of the form** $\neg(t \in D)$**,**

   (a) **t is an arithmetic term and** D **is of the form** $n1..n2$
   Let g_i is an unique label for G_i
   $\Pi_{prolog} := \Pi_{prolog} \cup g_i(t) : -n\# > n_2. \cup g_i(t) : -n\# < n_2.$
   $BODY := BODY \wedge g_i(t)$

   (b) **t is an arbitrary term, D is of the form** $\{t1, t2, \ldots, tn\}$
   $\Pi_{prolog} := \Pi_{prolog} \cup set\_d(X) : -member(X, [t1, t2, \ldots, tn]).$[2]
   $BODY := BODY \wedge \ \ \backslash + set\_d(t)$

[1]if t is arithmetic, we preprocess D and remove all non-numbers to avoid errors
[2]same as above

End for

Let $V_1, \ldots V_n$ be all arithmetic variables in $G_1, \ldots, G_n$

$BODY : -REORDER(BODY) \cap labeling([], [V_1, \ldots V_n])$.

$\Pi_{prolog} := \Pi_{prolog} \cap p : -BODY$.

If the answer to query ?-p to program Π_{prolog} is 'yes', return true; else return false.

END