Note: Recall that a Horn clause is a disjunction of literals with \textit{at most one} positive literal. A \textbf{purely negative} Horn clause contains only negative literals. A \textbf{normal} Horn clause contains zero or more negative literals and \textit{exactly one} positive literal. A \textbf{Horn formula} is a conjunction of Horn clauses.

In the following outline, the symbols $x_1, x_2, \ldots, x_n$ denote the Boolean variables used in the formula.

\textbf{Outline of the Algorithm:}

1. Set $x_i = \text{False}$, $1 \leq i \leq n$. (Thus, all variables are set to \text{FALSE} initially.)

2. \textbf{while} (there is an unsatisfied normal clause) \textbf{do}
   
   Change the assignment for the positive literal in the normal clause to \text{TRUE}.

3. \textbf{if} (all purely negative clauses are satisfied) \textbf{then}
   
   Return the assignment.

   \textbf{else} Output “No Satisfying assignment exists”.

\textbf{Lemmas Needed to Prove the Correctness of the Above Algorithm:}

\textbf{Lemma 1:} Suppose the above algorithm sets variables $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ to \text{TRUE} in that order in the loop of Step 2. Then, for each $j$, $1 \leq j \leq k$, $x_{i_j}$ must be set to \text{TRUE} in every satisfying assignment to the given Horn formula.

\textbf{Lemma 2:} When the above algorithm outputs “No”, there is no satisfying assignment to the given Horn formula.