1. Show that the INDEPENDENT-SET decision problem is NP-hard.

   Input: An undirected graph \( G = (V, E) \), positive integer \( k \).

   Question: Does \( G \) contain an independent set of size at least \( k \)? (Recall that an independent set is a subset of vertices with no edge between any two members of this subset.)

   **Solution:** First, choose decision problem \( D \) for reduction. How to choose? Pick problem “close” to IS, if possible, to make reduction easier. We choose VERTEX-COVER (VC) and show VC \( \leq_p \) IS:

   On input \((G, k)\) (for VC), construct \((G', k')\) (for IS) as follows:
   
   Set \( G' = G \) and \( k' = n - k \) (where \( n = |V| \) in \( G \)).

   Clearly, \((G', k')\) can be computed from \((G, k)\) in polytime. Also, if \( G \) contains a vertex cover \( C \) of size \( k \) or less, then \( V - C \) is an independent set in \( G \) of size \( n - k \) or more: since every edge of \( G \) has at least one endpoint in \( C \), no edge has both endpoints in \( V - C \). Finally, if \( G \) contains an independent set \( I \) of size \( n - k \) or more, then \( V - I \) is a vertex cover of size \( k \) or less: since no edge of \( G \) has both endpoints in \( I \), every edge of \( G \) has at least one endpoint in \( V - I \).

2. Show that CLIQUE is NP-hard

   Input: An undirected graph \( G = (V, E) \) and a positive integer \( k \).

   Question: Does \( G \) contain a clique of size at least \( k \), i.e., a subset of \( k \) or more vertices such that \( G \) contains every possible edge between the vertices in the clique?

   **Solution:** We show that CLIQUE is NP-hard by proving IS \( \leq_p \) CLIQUE (where IS is the INDEPENDENT-SET problem).

   On input \((G, k)\) (for IS), where \( G = (V, E) \), construct \((G', k')\) (for CLIQUE) as follows:
   
   Set \( k' = k \) and \( G' = (V, \bar{E}) \), where \( \bar{E} \) is the complement of \( E \), i.e., for all \( x, y \in V \), \((x, y) \in \bar{E} \iff (x, y) \notin E \).

   Clearly, \((G', k')\) can be computed from \((G, k)\) in polytime (in linear time, in fact).

   Also, if \( G \) contains an independent set \( I \) of size \( k \) or more, then \( I \) forms a clique in \( G' \): since \( G \) contains no edge between any two vertices of \( I \), \( G' \) contains every edge between any two vertices of \( I \).

   Finally, if \( G' \) contains a clique \( C \) of size \( k \) or more, then \( C \) forms an independent set in \( G \): since \( G' \) contains every edge between any two vertices of \( C \), \( G \) contains no edge between any two vertices of \( C \).

3. Show that LargeSAT is NP-hard.

   Input: Propositional formula \( F \) in CNF, positive integer \( k \)

   Question: Is there an assignment of values to the variables of \( F \) that makes at least \( k \) clauses of \( F \) True?

   **Solution:** CNF-SAT \( \leq_p \) LargeSAT

   Reduction:

   On input \( F \), output \((F, m)\) where \( m \) is the number of clauses in \( F \).

   Clearly, \((F, m)\) can be computed from \( F \) in polytime.

   Also, \( F \) is satisfiable iff there is an assignment of values to the variables of \( F \) that makes at least \( m \) clauses True (by definition of “satisfiable”).
4. Show that Restricted3SAT is NP-hard.

Input: Propositional formula $F$ in 3CNF, where no variable appears in more than three clauses.

Question: Is there an assignment of values to the variables of $F$ that makes $F$ True?

**Solution**: $3\text{SAT} \leq_p \text{Restricted3SAT}$

On input $F$, construct $F'$ as follows:

Start with $F' = F$.
Scan $F'$ and for each variable $x$ that appears in more than three clauses, replace the first occurrence of $x$ with a new variable $x_1$, the second occurrence with $x_2$, ..., the $k$-th occurrence with $x_k$. Then add clauses

$$(\sim x_1 \lor x_2 \lor x_2) \land (\sim x_2 \lor x_3 \lor x_3) \land ... \land (\sim x_{k-1} \lor x_k \lor x_k) \land (\sim x_k \lor x_1 \lor x_1)$$

$F'$ can be constructed from $F$ in polytime: the replacement of variables is done in linear time, and at most a linear number of new clauses need to be added.

Also, if $F$ is satisfiable, then it is possible to set the variables of $F'$ to match (make all the new variables have the same value as the old variable they correspond to), and this will satisfy $F'$.

Finally, if $F'$ is satisfiable, then every new variable corresponding to some old variable $x$ must be set to the same value (to make the new clauses True), so the old variable $x$ can be set to this value and doing this for each variable will satisfy $F$. 