1. Show that if $\Pi_1 \leq_p \Pi_2$ and Π_2 is NP-complete, does not imply that Π_1 is NP-complete. Give an example of two decision problems where
   - $\Pi_1 \in P$
   - $\Pi_2 \in$NP-complete
   - $\Pi_1 \leq_p \Pi_2$

   You must do the following:

   (a) For each problem Π_1, Π_2 clearly describe their input.

   (b) Give a polynomial time algorithm that solves Π_1.

   (c) Explain how to prove that $\Pi_2 \in$NP-complete.

   (d) **Prove** that $\Pi_1 \leq_p \Pi_2$

2. Given the points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$, we want to find the polynomial $P_n(x)$ of degree n that goes exactly through them. We know that given a point x, a polynomial that goes through the points $(x_i, y_i), \ldots, (x_j, y_j)$ is given by the following recursive formula

$$P_{i,j}(x) = \frac{(x_j - x)P_{i,j-1}(x) + (x - x_i)P_{i+1,j}(x)}{x_j - x_i}$$

   (a) Notice that $P_{k,k}$ is a polynomial of degree 0 (constant) that goes exactly through point (x_k, y_k). What is $P_{k,k}(x) =$

   (b) Provide a dynamic programming algorithm to compute $P_{0,n}(x)$

   (c) What is the time complexity of your algorithm?

3. Show that NP-complete languages are not closed under regular operations (union, concatenation, and kleene star).

4. **The Independent set problem:** Given a graph $G = (V, E)$, and an integer k. Is there a set of vertices $V' \subseteq V$ such that $\forall u, v \in V'$, $(u, v) \notin E$?

   Show that the *Independent Set Problem* is NP-hard by reducing it from Clique (HINT: use the complement of the graph)

5. Let IS5 be the Independent set problem restricted to graphs where the maximum degree is 5.

   (a) Show that IS5 is in NP

   (b) Give a polynomial time algorithm to solve IS5.

6. (Decision vs search problems) Given a polynomial time Turing machine M that on input (graph G, integer k) returns YES iff there is a vertex cover of size k.

   Use the given Turing Machine M to design a polynomial time algorithm that on (input graph G, integer k) returns all the vertices of a vertex cover of size k.