1. (25 Pts.) Suppose that someone gives you a polynomial-time algorithm to decide formula satisfiability (the decision problem). Describe how to use this algorithm to find satisfying assignments in polynomial time.

2. (25 Pts.) Prove that King Arthur’s problem is NP-complete. Here is a description of King Arthur’s problem: given \( n \) knights and the condition that some pairs of knights are enemies (the input maybe coded as the sequence of knights and pairs of enemies), is it possible to arrange the knights around a round table so that no pair of knights who are enemies sit side by side?

3. (25 Pts.) Prove that the vertex-cover problem remains NP-complete even if all the vertices in the graph are restricted to have even degree.

4. (25 pts.) Show that the following problem is NP-complete.
   \textbf{LongestPath}: given an undirected graph \( G \), and an integer \( L \), does there exist a simple path (between any two nodes) that contains at least \( L \) arcs? (Hint: Use Hamiltonian Cycle problem)

5. (25 pts.) \textbf{Extra Credit} The longest-simple-cycle problem is the problem of determining a simple cycle (no repeated vertices) of maximum length in a graph. Show that this problem is NP-complete.