1. Show that, if $P = NP$, a polynomial time algorithm exists that, given a satisfiable boolean formula $\Phi$, actually produces a satisfying assignment for $\Phi$. (Note: $NP$ is a class of languages, and producing an assignment for a formula is a function. Thus simply saying that "because SAT is in $NP$, we are done" is not enough.)

2. **Knapsack** is the following problem: Given integers $K$ and $B$, and items $\{1, 2, \ldots, n\}$ each with integer size $s_i$ and integer profit $p_i$, find a subset $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} s_i \leq B$ and $\sum_{i \in S} p_i \geq K$. The common optimization problem is: we attempt to maximize $K$, the total profit, with a bound $B$ on the total size of items one can pick.

   Formulate **Knapsack** as a language and prove the language is NP-Complete. The reduction must come from a problem claimed NP-Complete in the notes (use all the theorems from the notes even if we did not prove it yet in class).

3. Consider the following problem, called **SHORTEST SIMPLE $s - t$ PATH**: Given a directed graph $G = (V, E, w)$, where $w(e)$ is defined as a (possibly negative) integer for each edge $e \in E$, vertices $s, t \in V$, and a positive integer $K$, answer YES if there is a simple $s - t$ path of total weight at most $K$. An $s - t$ path starts at $s$ and ends at $t$, a path is simple if it does not repeat any vertices, and the total weight of a path is the sum of the weights of its edges.

   Formulate the **SHORTEST SIMPLE $s - t$ PATH** problem as a language $L_S$. Then prove that $L_S$ is NP-complete. Simplest reduction is from HAMPATH.