1. Prove that the following problems are self reducible by a (direct) polynomial Cook reduction from the search version to the decision version of the same problem.

   (a) $\mathsf{Clique} = \{ (G,k) \mid G \text{ contains a clique of size } k \}$.[1]

   (b) $\mathsf{GraphIsomorphism} = \{ (G_1, G_2) \mid G_1 \text{ and } G_2 \text{ are isomorphic} \}$.[2]

2. For a number $n \in \mathbb{N}$, denote by $bin(n)$ the binary representation of n, e.g., $bin(13) = 1101$. Let $L \subseteq \{1\}^*$ be a unary language, and define $bin(L) = \{bin(n) \mid 1^n \in L\}$. Show that $L \in \mathsf{P}$ if and only if $bin(L) \in \mathsf{E}$, where $\mathsf{E} = \bigcup_{c \geq 1} \mathsf{DTIME}(2^{cn})$.

3. Let UpToOneSat be the following language:
   $\mathsf{UpToOneSat} = \{ \phi \mid \phi \text{ is a CNF formula that has at most one satisfying assignment}\}$.
   Prove that $\mathsf{UpToOneSat} \in \mathsf{NP}$ if and only if $\mathsf{NP} = \mathsf{coNP}$.

4. We say that a non-deterministic machine is *nice* if for every input $x \in \{0,1\}^*$ the following holds: every computation path returns either 'accept','reject' or 'quit'. There is at least one non-quit path, and all non-quit paths have the same value. Let NICE be the class of all languages that are accepted by some non-deterministic, polynomial time, nice machine. Prove that $\mathsf{NICE} = \mathsf{NP} \cap \mathsf{coNP}$.

5. The class DP is defined as the set of all languages L for which there are two languages $L_1 \in \mathsf{NP}$ and $L_2 \in \mathsf{coNP}$ such that $L = L_1 \cap L_2$. Let SAT-UNSAT be the language of all the pairs (ϕ_1, ϕ_2) such that ϕ_1 and ϕ_2 are CNF formulas, ϕ_1 is satisfiable and ϕ_2 is not. Show that SAT-UNSAT is DP-complete, i.e., $\mathsf{SAT\text{-}UNSAT} \in \mathsf{DP}$ and every language in DP is polynomial-time reducible to it.

[1] The decision version is "Given a pair (G,k) does G contain a clique of size k?" and the search version is "Given a pair (G,k) find a clique of size k in G if exists, and reject otherwise".

[2] Two graphs are *isomorphic* if there is a way to label the vertices of one graph, such that the two graphs become identical.