The purpose of this problem set is to reinforce the definitions of $NP_{\text{search}}$, $P_{\text{search}}$, and $NP_{\text{search}}$-completeness and practice $NP$-completeness proofs. Note that the new **Sunday deadline** (due to the FAS policy that problem sets cannot be due after the first three days of reading period). Because of this change, this is a half-length problem set and we are also giving you 2 extra late days that you can use on this problem set (in addition to using up to 3 of your original late days that you have left over).

1. (Reductions to and from easy problems)
   
   (a) Prove that if a problem $\Pi$ is in $P_{\text{search}}$, then $\Pi \leq_p \Gamma$ for all computational problems $\Gamma$.
   
   (b) Show that all problems in $NP_{\text{search}}$ are $NP_{\text{search}}$-complete if and only if $NP_{\text{search}} \subseteq P_{\text{search}}$ (equivalently, $P = NP$).

2. (Monotone SAT) A boolean formula is *monotone* if there are no negations in it. Restricting SAT to Monotone formulas makes it trivial; setting all variables to 1 is always a satisfying assignment.

   However, the following variant of Monotone (3-)SAT is more interesting:

   | **Input** | A monotone (3-)CNF formula $\varphi(x_0, \ldots, x_{n-1})$ and a number $k \in \mathbb{N}$ |
   | **Output** | A satisfying assignment $\alpha \in \{0, 1\}^n$ in which at least $k$ variables are set to 0, or $\bot$ if no such $\alpha$ exists |

   **Computational Problem** $k$-False Monotone (3-)SAT

   (a) Prove that $k$-False Monotone 3-SAT is $NP_{\text{search}}$-complete, even when $k = n/2$. (Hint: reduce from 3-SAT, replacing negated variables with new ones and adding additional clauses.)

   (b) Show that if we fix $k = 3$, then $k$-False Monotone SAT is in $P$. (Hint: show that it suffices to consider assignments in which exactly 3 variables are set to 0.)

   (c) (*challenge) Show that $k$-False Monotone 2-SAT is $NP_{\text{search}}$-complete. (Hint: reduce from Independent Set.)