1. Suppose you are given a magic black box that somehow answers the following decision problem in polynomial time:
   • **INPUT:** A boolean circuit \( K \) with \( n \) inputs and one output.
   • **OUTPUT:** True if there are input values \( x_1, x_2, \ldots, x_n \in \{\text{True, False}\} \) that make \( K \) output True, and False otherwise.

Using this black box as a subroutine, describe an algorithm that solves the following related search problem in polynomial time:
   • **INPUT:** A boolean circuit \( K \) with \( n \) inputs and one output.
   • **OUTPUT:** Input values \( x_1, x_2, \ldots, x_n \in \{\text{True, False}\} \) that make \( K \) output True, or None if there are no such inputs.

   [Hint: You can use the magic box more than once.]

2. An **independent set** in a graph \( G \) is a subset \( S \) of the vertices of \( G \), such that no two vertices in \( S \) are connected by an edge in \( G \). Suppose you are given a magic black box that somehow answers the following decision problem in polynomial time:
   • **INPUT:** An undirected graph \( G \) and an integer \( k \).
   • **OUTPUT:** True if \( G \) has an independent set of size \( k \), and False otherwise.

   (a) Using this black box as a subroutine, describe algorithms that solves the following optimization problem in polynomial time:
      • **INPUT:** An undirected graph \( G \).
      • **OUTPUT:** The size of the largest independent set in \( G \).

      [Hint: You’ve seen this problem before.]

   (b) Using this black box as a subroutine, describe algorithms that solves the following search problem in polynomial time:
      • **INPUT:** An undirected graph \( G \).
      • **OUTPUT:** An independent set in \( G \) of maximum size.
To think about later:

3. Formally, a **proper coloring** of a graph $G = (V, E)$ is a function $c: V \rightarrow \{1, 2, \ldots, k\}$, for some integer $k$, such that $c(u) \neq c(v)$ for all $uv \in E$. Less formally, a valid coloring assigns each vertex of $G$ a color, such that every edge in $G$ has endpoints with different colors. The **chromatic number** of a graph is the minimum number of colors in a proper coloring of $G$.

   Suppose you are given a magic black box that somehow answers the following decision problem in polynomial time:
   - **INPUT**: An undirected graph $G$ and an integer $k$.
   - **OUTPUT**: TRUE if $G$ has a proper coloring with $k$ colors, and FALSE otherwise.

Using this black box as a subroutine, describe an algorithm that solves the following coloring problem in polynomial time:
   - **INPUT**: An undirected graph $G$.
   - **OUTPUT**: A valid coloring of $G$ using the minimum possible number of colors.

   [Hint: You can use the magic box more than once. The input to the magic box is a graph and only a graph, meaning only vertices and edges.]