1a) We will show that the problem 2-COLORABLE is in P by reducing it to 2-SAT, which is known to be in P. Given a graph G, we will label each vertex as $v_1, v_2, \dots v_n$ and each edge as (v_i, v_j). For each vertex, we will have $x_1, x_2, \dots x_n$ as a boolean (TRUE or FALSE) that represents the color assignment. For every edge (v_i, v_j) we have two clauses: 1. $(x_i \vee x_j)$ and 2. $(\neg x_i \vee \neg x_j)$. These clauses make sure that any valid truth assignment causes the adjacent vertices to have different colors so one variable is TRUE and the other is FALSE. Since the number of vertices and edges in G is polynomial in the size of the input, the reduction runs in polynomial time. If G is 2-colorable, we can assign the variables | one color corresponds to TRUE and the other corresponds to FALSE. Since no two adjacent vertices have the same color because the only way to not satisfy would be if an edge's endpoints had the same color, the corresponding 2-SAT is satisfiable. ∴ "yes" maps to "yes." If the 2-SAT is satisfiable, there is a truth assignment that satisfies the clauses. This ensures that adjacent vertices in G have different truth values, so we can construct a valid 2-coloring. If the 2-SAT is satisfiable, then G is 2-colorable. ∴ "no" maps to "no." Since 2-SAT is known to be solvable in polynomial time and we have reduced 2-COLORABLE to 2-SAT, then 2-COLORABLE is in P.

b) G can be verified in polynomial time by checking that every edge has endpoints with distinct colors. ∴ 3-COLORABLE is in NP. We will show that 3-COLORABLE is NP-hard with a reduction from 3-SAT. Consider the reduction with ϕ that produces G where we have 3 special vertices, X, Y, Z, and 1 vertex for each literal, $a_i, \bar{a}_i$. Consider a triangle on X, a_i, $\bar{a}_i$ for each i. Since any graph that is 3-COLORABLE must have different colors for Y and Z, we can let the color assigned to Y be TRUE and the color for Z be FALSE. Similarly, every a_i and $\bar{a}_i$ should be colored with TRUE and FALSE or FALSE and TRUE. For each clause $(a \vee b \vee c)$ in ϕ consider the graph that was given. Let the three grey nodes on the left represent a, b, c and the grey node on the right be Y | this reduction runs in polynomial time. Note that if we start with a yes instance of 3-SAT then we can show that the production yields a yes instance from 3-COLORABLE. Consider a satisfied instance of ϕ such that we assign X a random color, Y as TRUE, and Z as FALSE such that if a_i is true then we assign a_i TRUE and $\bar{a}_i$ FALSE and vice versa if a_i is false. Also, note that every clause of the graph has at least one node labeled TRUE among the 3 grey nodes on the left | the grey node on the right must also be assigned TRUE. Given that if each of the grey nodes are colored with one of two colors, then we can extend this coloring to a 3-coloring iff at least one of the three grey nodes on the left has the same color as the one on the right, we can extend this 3-coloring to the entire graph.

b) Note that if G is a yes instance of 3-COLORABLE then we can see that the reduction has a satisfied ϕ. Consider a 3-coloring of G such that x, y, z are assigned distinct colors and y is assigned TRUE and z is assigned FALSE such that each pair $a_i, \overline{a_i}$ are either TRUE and FALSE or FALSE and TRUE. For each clause, consider the graph | the right grey node must be assigned TRUE and thus the only way for it to be 3-colored is if at least one of its left grey nodes is TRUE. This means we can assign a_i to true if it is assigned TRUE and $\overline{a_i}$ to false if it is assigned FALSE | this will satisfy every clause. $\therefore$ a satisfied ϕ exists. Therefore, 3-COLORABLE is NP-complete.

2) First, note that (3, 3)-SAT is in NP because 3-SAT is in NP. Now we will show that every A in NP is polynomial reducible to (3,3)-SAT. Consider a 3-CNF formula ϕ where we can perform the following transitions to obtain a formula ϕ'. Given a_i, we will replace the n_i^{th} occurrences of it with the variables $b_{i,1}, b_{i,2}, b_{i,3} \cdots, b_{i,n_i}$ and we add clauses $(\overline{b_{i,1}} \vee b_{i,2}), (\overline{b_{i,2}} \vee b_{i,3}), (\overline{b_{i,n_i-1}} \vee b_{i,n_i}), (\overline{b_{i,n_i}} \vee b_{i,1})$ such that any assignment that satisfies ϕ' is setting all these variables to the same value. This shows that "yes" maps to "yes" and "no" maps to "no" since any satisfied ϕ can become a satisfied ϕ' by setting $a_i = b_{i,1}$ and any satisfied ϕ can become a satisfied ϕ' by setting $b_{i,j} = x_i$ for all i and j. $\therefore$ (3, 3)-SAT is NP-complete.

3) We will prove that MAX2SAT is NP-complete by reducing from 3-SAT. Given a 2-CNF formula ϕ and an integer k, verifying whether a truth assignment satisfies at least k clauses can be done in polynomial time. To prove MAX2SAT is NP-hard, we construct a reduction from 3-SAT. Given a 3-SAT formula ϕ with m clauses of the form $(a \vee b \vee c)$, we transform it into an equivalent MAX2SAT instance as follows: for each clause $(a \vee b \vee c)$, we introduce 10 specific 2-literal clauses (the ones in the hint). A fresh variable w is introduced for each clause. The resulting formula ϕ' has $10m$ clauses and we'll set the threshold as $k=7m$. The transformation runs in polynomial time. If $x, y,$ and z are false, setting $w=$ true satisfies 4 clauses, $w=$ false satisfies 6 clauses, so maximum 6 clauses can be satisfied. If one of x, y, z is true, setting $w=$ false will satisfy 7 clauses. If two of x, y, z are true, setting $w=$ true will satisfy 7 clauses. If $x, y,$ and z are all true, setting $w=$ true satisfies 7 clauses. No assignment satisfies more than 7 clauses with any group of 10. If ϕ is satisfiable, then MAX2SAT has an assignment with $\geq 7m$ satisfied clauses. We will assign truth values to x, y, z in each group based on a satisfying assignment of ϕ. From the $k=7m$ proof, we can extend this assignment to w so that exactly 7 of 10 clauses in each group are satisfied. Since there are m clauses in ϕ, this ensures that $\geq 7m$ clauses in the MAX2SAT instance are satisfied. ∴ yes maps to yes. If $\geq 7m$ clauses of MAX2SAT are satisfied, then ϕ is satisfiable. Each group of 10 clauses can satisfy maximum 7 clauses. If $\geq 7m$ clauses are satisfied in total, the group must have exactly 7 satisfied clauses. From the $k=7m$ proof, this is only possible if in every clause gadget, one or more of x, y, z is true. ∴ the assignment that satisfies $7m$ clauses corresponds to a satisfying assignment for ϕ. Since this reduction runs in polynomial time and MAX2SAT is in NP, MAX2SAT is NP-complete.