1. (20 points) Let HALT be the Halting language. Show that HALT is NP-hard. Is it NP-complete?

2. (20 points) Show that, if P = NP, then every language \( A \in P \), except \( A = \emptyset \) and \( A = \Sigma^* \), is NP-complete. Here \( \Sigma \) is the alphabet, and you may assume that it is \{0, 1\}.

3. (20 points) Let \( \phi \) be a 3CNF. An \( \neq \)-assignment to the variables of \( \phi \) is one where each clause contains two literals with unequal truth values.
   
   (a) Show that any \( \neq \)-assignment automatically satisfies \( \phi \), and the negation of any \( \neq \)-assignment to \( \phi \) is also an \( \neq \)-assignment.
   
   (b) Let \( \neq SAT \) be the collection of 3CNFs that have an \( \neq \)-assignment. Show that we obtain a polynomial time reduction from 3SAT to \( \neq SAT \) by replacing each clause
   
   \[ c_i = (y_1 \lor y_2 \lor y_3) \]
   
   with the two clauses
   
   \[ (y_1 \lor y_2 \lor z_i) \quad \text{and} \quad (\overline{z}_i \lor y_3 \lor b), \]
   
   where \( z_i \) is a new variable for each clause \( c_i \) and \( b \) is a single additional new variable.

   (c) Conclude that \( \neq SAT \) is NP-complete.

4. (20 points) Let DOUBLE-SAT = \{\( \phi | \phi \) is a CNF that has at least two satisfying assignments\}. Show that DOUBLE-SAT is NP-complete.

5. (20 points) A subset of the nodes of a graph \( G \) is a dominating set if every other node of \( G \) is adjacent to some node in the subset. Let

   \[ \text{DOMINATING-SET} = \{ (G, k) | G \text{ has a dominating set with } k \text{ nodes} \}. \]

   Show that it is NP-complete by giving a reduction from VERTEX-COVER. You can assume that \( G \) has no vertex with degree 0.