1. Let CFL-Int-Reg = CFL-IR = \{⟨G,M⟩ | G is a context free grammar, M is a DFA 
and \(L(G) \cap L(M) \neq \emptyset\}\).
Show that CFL-IR is decidable.
Hint. Use a reduction to Empty-CFL.

2. Let \(L(x)\) be a computable function that lists always halting integer output programs:
a program is an always halting integer output program if for every possible input it 
eventually outputs an integer.
So \(L\) is computed by a program, \(P_L\) say. Specifically, \(L(1), L(2), \cdots = P_{j_1}, P_{j_2}, \cdots\) is a 
list of always halting integer output programs.
Prove, by diagonalization, that there is an always halting integer output program \(Q\) 
not on the list generated by \(L\).

3. Let \(A\) and \(B\) be recursively enumerable sets. Suppose that \(\overline{A} \cap \overline{B} = \emptyset\). Show that 
there is a decidable set \(C\) such that \(\overline{A} \subseteq C\) and \(\overline{B} \subseteq \overline{C}\).
Hint. Modify the algorithm for deciding membership in set \(L\) if both \(L\) and \(\overline{L}\) are 
recursively enumerable. (\(A\) and \(B\) will replace \(L\) and \(\overline{L}\); you will need to define \(L\) 
suitably. This will involve three cases for your membership test: when \(x \in A - B\), 
when \(x \in B - A\), and when \(x \in A \cap B\).)

4. Let Halt-Exactly-Once = HEO = \{⟨Q⟩ | Q halts on exactly one input\}.
Suppose that you are given an algorithm \(A_{\text{HEO}}\) that decides HEO. Using \(A_{\text{HEO}}\) as a 
subroutine, give an algorithm \(A_{\text{H}}\) to decide \(H\).

5. Let Mixed = \{⟨Q⟩ | Q halts on input 1 and does not halt on input 2\}.
Suppose that you are given an algorithm \(A_{\text{Mixed}}\) that decides Mixed. Using \(A_{\text{Mixed}}\) as a 
subroutine, give an algorithm \(A_{\text{H}}\) to decide \(H\).