1. Let \( V = \{ \langle P \rangle \mid P \text{ computes the identity function } P(\langle x \rangle) = x \text{ for all } x \} \).
   Suppose you are given an algorithm \( A_V \) that decides \( V \). Using \( A_V \) as a subroutine, give an algorithm \( A_{\text{prog}} \) to decide \( A_{\text{prog}} \).

2. Let \( W = \{ \langle P \rangle \mid P \text{ halts on input } 0 \text{ and does not halt on input } 1 \} \).
   Suppose you are given an algorithm \( A_W \) that decides \( W \). Using \( A_W \) as a subroutine, give an algorithm \( A_{\text{prog}} \) to decide \( A_{\text{prog}} \).

3. a. Let \( X = \{ \langle P \rangle \mid P \text{ contains a line of code that is not executed on input } 1 \} \).
   Suppose you are given an algorithm \( A_X \) that decides \( X \). Using \( A_X \) as a subroutine, give an algorithm \( A_{\text{prog}} \) to decide \( A_{\text{prog}} \).

   b. Let \( Y = \{ \langle P \rangle \mid P \text{ contains a line of code that is not executed on any input} \} \).
   Suppose you are given an algorithm \( A_Y \) that decides \( Y \). Using \( A_Y \) as a subroutine, give an algorithm \( A_{\text{prog}} \) to decide \( A_{\text{prog}} \).

4. Let \( Z = \{ \langle P \rangle \mid P \text{ contains a variable that is never assigned a value on any input} \} \).
   Suppose you are given an algorithm \( A_Z \) that decides \( Z \). Using \( A_Z \) as a subroutine, give an algorithm \( A_{\text{prog}} \) to decide \( A_{\text{prog}} \), or to decide some other undecidable language encountered in class.