1. Let $Sometime-Halt = \{\langle Q \rangle \mid Q \text{ halts on at least one input}\}$.
   
   Suppose that you are given an algorithm $A_{Sometime-Halt}$ that decides $Sometime-Halt$. Using $A_{Sometime-Halt}$ as a subroutine, give an algorithm $A_H$ to decide $H$.

2. Let $Mixed = \{\langle Q \rangle \mid Q \text{ halts on input 0 and does not halt on input 1}\}$.
   
   Suppose that you are given an algorithm $A_{Mixed}$ that decides $Mixed$. Using $A_{Mixed}$ as a subroutine, give an algorithm $A_H$ to decide $H$.

3. Let $Useless-Var = \{\langle Q \rangle \mid Q \text{ contains a variable that remains zero whatever the input to } Q\}$.
   
   Suppose that you are given an algorithm $A_{Useless-Var}$ that decides $Useless-Var$. Using $A_{Useless-Var}$ as a subroutine, give an algorithm $A_H$ to decide $H$.

4. Let $Inf-Halt = \{\langle Q \rangle \mid Q \text{ halts on infinitely many inputs}\}$.
   
   Suppose that you are given an algorithm $A_{Inf-Halt}$ that decides $Inf-Halt$. Using $A_{Inf-Halt}$ as a subroutine, give an algorithm $A_H$ to decide $H$, or give an algorithm to decide some other undecidable language encountered on the handout Undecidability, Part 2.

5. Let $Inf-Not-Halt = \{\langle Q \rangle \mid Q \text{ fails to halt on infinitely many inputs}\}$.
   
   Suppose that you are given an algorithm $A_{Inf-Not-Halt}$ that decides $Inf-Not-Halt$. Using $A_{Inf-Not-Halt}$ as a subroutine, give an algorithm $A_H$ to decide $H$, or give an algorithm to decide some other undecidable language encountered on the handout Undecidability, Part 2.