1. Describe an appropriate reduction to show that the following function is not computable, where `P` is any program that takes exactly one input `x`. Don't forget to argue that your reduction is correct!

$$\texttt{allstop(P)} = \begin{cases} \text{True} & \text{if } \texttt{P(x)} \text{ halts for every input } \texttt{x}, \\ \text{False} & \text{otherwise.} \end{cases}$$

For a contradiction, assume that `allstop(P)` is computable.

Consider the following program.

```
def h(P,x):
    def P1(y):
        return P(x)

    return allstop(P1)
```

Then for all programs `P` and inputs `x`:

`h(P,x)` returns True if `allstop(P1)` returns True
if `P1(y)` halts for *every* input `y`
if `P(x)` halts;

`h(P,x)` returns False if `allstop(P1)` returns False
if `P1(y)` does not halt for *some* input `y`
if `P(x)` does not halt.

(This works because `P1(y)` halts for all inputs `y` iff `P(x)` halts.)

In other words, `h(P,x)` computes function `halt`, a contradiction!

Hence, by contradiction, `some` is not computable.

2. Describe an appropriate reduction to show that the following function is not computable, where P is any program that takes exactly one input x. Don't forget to argue that your reduction is correct!

$$\texttt{steps(P,x)} = \begin{cases} \text{the number of lines of code executed by } \texttt{P} \text{ on input } \texttt{x}, & \text{if } \texttt{P(x)} \text{ halts,} \\ 0 & \text{otherwise.} \end{cases}$$

For a contradiction, assume that `steps(P,x)` is computable.

Consider the following program.

```
def h(P,x):
        return steps(P,x) > 0
```

Then for all programs `P` and inputs `x`:

`h(P,x)` returns True if `steps(P,x)` returns a positive integer if `P(x)` halts;

`h(P,x)` returns False if `steps(P,x)` returns 0 if `P(x)` does not halt.

In other words, `h(P,x)` computes function `halt`, a contradiction!

Hence, by contradiction, `steps` is not computable.