Q1 Computability
5 Points

Q1.1 Halting Problem
1 Point

There exists no program that can solve the Halting Problem.

- (●) True.
- ( ) False.

Explanation

See Note 12.

Q1.2 TestHalt
1 Point

Which of the following correctly describes the program $\texttt{TestHalt}(P, x)$?

- (●) $\texttt{TestHalt}(P, x)$ returns **True** if program **P** halts, when it is run on input x, and returns **False** otherwise.
- ( ) $\texttt{TestHalt}(P, x)$ returns **False** if program **P** halts, when it is run on input x, and returns **True** otherwise.

Explanation

See Note 12.

Q1.3
1 Point

Suppose you want to prove that a program P is uncomputable, which one of the following argument would work?

- ( ) Show that P can be computed if the halting problem can be solved.
- (•) Show that the halting problem can be solved if P can be computed.

Explanation

The first option can be restated as *(the halting problem can be solved)* $\Rightarrow$ *(P can be computed)*. The halting problem cannot be solved, so this is vacuously true. On the other hand, the second option tells us that if we can compute P, then we can solve the halting problem. Solving the halting problem is impossible, so P must be uncomputable.

Q1.4
1 Point

Find the computable problem from the following problems.

- ( ) Determining whether a given program P halts when given an empty input
- (•) Determining whether a given program P halts in 10^{100} steps when given an empty input.
- ( ) Determining whether a given program P on input x prints anything during the execution.

Explanation

- **We present the following reduction. Let $Q(x)$ be the halting problem taking input x. We define program P that on input y executes $Q(x)$ oblivious to y. Then P halts on the empty input if and only if $Q(x)$ halts.**
- **We can just run $P(x)$ for the first 10^{100} steps and see if it halts. This is not practical, but it is still computable.**
- **We present the following reduction. Let $Q(x)$ be the halting problem taking input x. We define program P that on input x executes $Q(x)$ and then output a string `halt`. Then P outputs if and only if $Q(x)$ halts.**

Q1.5
1 Point

You are given a program FlipHalt that takes in a program P and input x. It returns True if P does not halt on x, and returns False if P does halt on x. Which of the following is true?

- ( ) FlipHalt is a correct solution to the halting problem.
- (•) FlipHalt cannot exist, because it would also solve the halting problem.
- ( ) FlipHalt can be used to detect infinite loops, but not halting programs.

Explanation

If FlipHalt existed, we could use it to construct TestHalt by negating its output. Since the halting problem is uncomputable, FlipHalt must also be uncomputable. This is a classic example of a reduction — solving FlipHalt would solve the halting problem.