1. This problem is recognizable but undecidable.

**Proof it’s recognizable**: we can write a program that, when given input , simulates on input and reads the state of on every step, seeing what the value of is. If in any of the steps, then return true.

**Proof it’s undecidable**: Let’s make this problem stronger. Let’s define . State for this problem is . Assume this problem is decidable, then there exists a Turing machine that, when given a Turing machine and its input , will accept if and only if will reach state on input . We will reduce from the accepting problem, which we recall to be . Let’s make a new Turing machine that converts input into , where is the accept state of , which we define to be . simulates on input . We see that is a decider for the accepting problem. But we know that the accepting problem is undecidable. This contradiction shows our assumption of the existence of is wrong, which means this problem is undecidable.

1. recognizable, undecidable (reduce from halting problem)
2. unrecognizable (reduce from co-halting problem)
3. decidable (just brute force it)