Let $x$ range over a set $X$ of identifiers, $n$ range over the natural numbers $\mathbb{N}$, and $s$ range over stores: total functions from $X$ to $\mathbb{N}$.

Consider a language with the following abstract syntax.

$$ e ::= n \mid x := e \mid !x \mid e_1; e_2 $$

(a) Define a conventional deterministic small-step operational semantics $(e, s) \rightarrow (e', s')$ for the language. Comment briefly on the choices you make. [5 marks]

(b) If your language is deterministic and terminating, the operational semantics implicitly defines a more abstract semantics: we can regard each expression as a function over stores $\llbracket e \rrbracket$ that takes store $s$ to the unique number $n$ and store $s'$ such that

$$ (e, s) \rightarrow^* (n, s') \land \nexists e'', s''. (n, s') \rightarrow (e'', s'') $$

This language is quite limited in expressiveness. Describe, as clearly and precisely as you can, the set of functions from stores to (number, store) pairs that are expressible as $\llbracket e \rrbracket$ for some $e$. [5 marks]

(c) The primitive contexts $C$ for this language are expressions with a single hole:

$$ C ::= x := \_ \mid e_1; \_ \mid \_; e_2 $$

Write $C[e]$ for the expression resulting from replacing the hole in $C$ by $e$.

Say a binary relation $\sim$ over expressions is a congruence if $e \sim e'$ implies $\forall C. C[e] \sim C[e']$.

Say a binary relation $\sim$ over expressions respects final values if $e \sim e'$ implies $\forall s_0, n, n', s, s'. ((e, s_0) \rightarrow (n, s) \land (e', s_0) \rightarrow (n', s')) \Rightarrow n = n'$.

Use your characterisation of part (b) to define an equivalence relation over expressions that is a congruence and respects final values. Explain briefly why it has those properties. [4 marks]

(d) Define a terminating algorithm that, for any expressions $e$ and $e'$, computes whether $e \sim e'$ or not. Explain informally why it is correct. *Hint:* you may want to adapt your semantics from part (a) to compute symbolically. [6 marks]