Consider the language with functions, integers, and printing.

\[
\begin{align*}
\tau & ::= \text{unit} \mid \text{int} \mid \tau \rightarrow \tau' \\
e & ::= x \mid \lambda x : \tau. e \mid e \ e' \mid \text{skip} \mid n \mid \text{print}(e) \mid e; e'
\end{align*}
\]

The typing rule for `\text{print}(e)` is:

\[
\begin{array}{c}
\Gamma \vdash e : \text{int} \\
\Gamma \vdash \text{print}(e) : \text{unit}
\end{array}
\]

(a) Define a small-step, call-by-value operational semantics for this language. Clearly explain what the components of the machine configuration are, and how it identifies what is printed. [10 marks]

(b) State a progress theorem for this language, and explain what it says about the evolution of the machine state. [4 marks]

(c) Prove progress for the `\text{print}(e)` case, giving the names of any of the standard properties (such as substitution) that you needed to use in the proof. [6 marks]