I) 1. \((a|b)^*ab\)

2. \(a, b\) \hspace{1cm} \text{(or as in 3.)}

3. 

II) 4. \(Y, X, S\)

5. \(\text{Last}(S) = \{c, d\}\)
   \(\text{Last}(X) = \{c, d\}\)
   \(\text{Last}(Y) = \{d\}\)
   \(\text{Last}(cX) = \{c, d\}\)
   \(\text{Last}(dY) = \{d\}\)

6. Right recursion: \(X \rightarrow cX\) or \(Y \rightarrow dY\)

7. \(d : \)

\( \)
8. A) \[ S \rightarrow XY \quad \text{(or) \quad S \rightarrow Sd \quad \text{or other equivalent grammars)} \]
\[ X \rightarrow Xc \]
\[ X \rightarrow \varepsilon \]
\[ Y \rightarrow Yd \]
\[ Y \rightarrow \varepsilon \]

B) \[ c^*d^* \]

III) 9. \[ (p \& \& q) \]

10. 

11. \[ b_2 = \text{shortcircuit}(c_2, f, t); \]
\[ b_1 = \text{shortcircuit}(c_1, b_2, t); \]
\[ \text{return } b_1; \]
IV) 12. A) b
     B) a+b
     C) a+b

13. It did not preserve the value in %rbx.

14. A) No: The caller has to push all registers (that it cares about) to the stack before invoking the callee. This process happens n times whenever g() is called. It is hard to optimize g() with registers.

   Yes: We can make f() inline and there would be no harm to the performance of g().

   B) No. The callee cannot both update this register with the return value and keep it the same as before entering callee.