A. Let \( D = 0 \cup 1 \cup 2 \cup 3 \cup 4 \cup 5 \cup 6 \cup 7 \cup 8 \cup 9 \). Let
\[
Z = DDDD \cup DDDD - DDDD.
\]

B. Let \( A \) be a regular expression that matches all single upper-case letters, lower-case letters, and spaces \( _{..} \). Let
\[
S = DD^* _{..} AA^*.
\]

C. Let
\[
P = PO _{..} Box _{..} DD^*.
\]

D. Let \( C \) be a regular expression that matches all single upper-case letters. Let \( N \) be a regular expression that matches the newline and carriage return characters. Then the regular expression that we desire is
\[
AA^* N (S \cup P) NAA^* _{..} CC^* _{..} Z.
\]

[This problem is somewhat under-specified and open-ended. In grading, I am looking for reasonable answers that demonstrate basic competence with regular expressions. In other words, a perfect answer is not required. Just about any answer can be improved to a slightly better answer that handles more obscure cases.]

B. Let \( A \) be regular and \( B \) be context-free. Let \( M \) be a DFA for \( A \) and \( N \) a PDA for \( B \). We will design a PDA \( P \) for \( A \cap B \), that simulates \( M \) and \( N \) simultaneously and accepts if and only if both \( M \) and \( N \) accept. The stack of \( P \) will be used to simulate the stack of \( N \). Precisely, let
\[
\begin{align*}
\Sigma^P &= \Sigma^M = \Sigma^N, \\
\Gamma^P &= \Gamma^N, \\
Q^P &= Q^M \times Q^N, \\
q^P_0 &= (q^M_0, q^N_0), \text{ and} \\
F^P &= F^M \times F^N.
\end{align*}
\]

It remains to describe \( \delta^P \). For every transition \( \delta^M (q^M, a) = r^M \) and \( \delta^N (q^N, a, t) = (r^N, u) \), add a transition
\[
\delta^P ((q^M, q^N), a, t) = ((r^M, r^N), u).
\]

By our usual reasoning for the product construction, \( P \) accepts exactly \( A \cap B \).

C. [This is 1.49b in our textbook. By the way, 1.49a is more interesting.] Let \( A = \{1^n w : n \geq 0 \text{ and } w \text{ contains at most } n \text{ 1s} \} \subseteq \{0,1\}^* \). Assume for the sake of contradiction that \( A \) is
Let $p$ be the pumping length for $A$. Let $s = 1^p01^p$. Then $s \in A$ and $|s| \geq p$. By the pumping lemma, $s = xyz$ where $y \neq \epsilon$, $|xy| \leq p$, and $xy^iz \in A$ for all $i \geq 0$. It is easy to see that $xy$ is a substring of the first $1^p$ in $s$. Thus $y = 1^k$ for some $1 \leq k \leq p$, and $xy^0z = 1^{p-k}01^p$. When $1^{p-k}01^p$ is written in the form $1^n w$, it must be true that $n \leq p - k < p$ and there are at least $p$ 1s in $w$. Thus $xy^0z \notin A$. This contradiction implies that $A$ is not regular after all.

D. [This is 1.63a in our textbook.] Let $A$ be infinite and regular. Because $A$ is regular, there exists a pumping length $p$ for $A$. Because $A$ is infinite, there exists a string $s \in A$ such that $|s| \geq p$. By the pumping lemma, there exist strings $x, y, z$ such that $y \neq \epsilon$ and $xy^iz \in A$ for all $i \geq 0$. Let $B = \{xy^iz : i \text{ is even}\}$. Because $y \neq \epsilon$, $B$ is infinite. Because $x(yy)^*z$ is a regular expression for $B$, $B$ is regular. Let $C = A - B = A \cap \overline{B}$. Because $B$ is regular, so is $\overline{B}$. Because $A$ and $\overline{B}$ are regular, so is their intersection, which is $C$. Because $C$ contains $xy^iz$ for all odd $i$, $C$ is infinite. Finally, $B$ and $C$ are disjoint, and $B \cup C = A$. Thus $A$ is a disjoint union of two infinite, regular languages $B$ and $C$.

E. [This is 2.9 in our textbook.] This context-free grammar works for the given language:

$$S \rightarrow TC \mid AU, \quad C \rightarrow \epsilon cC, \quad A \rightarrow \epsilon aA, \quad T \rightarrow \epsilon aTb, \quad U \rightarrow \epsilon bUc.$$