A possible way to do it is the following:
Let $P = (\{p, q\}, \{0, 1\}, \{X, Z\}, \delta, p, Z_0)$
\[
\begin{align*}
\delta(p, 0, \alpha) &= \{(p, XX\alpha), (p, X\alpha)\} & \text{add 1 or 2 X for every 0} \\
\delta(p, \varepsilon, \alpha) &= \{(p, \alpha)\} & \text{done with the 0} \\
\delta(q, 1, X\alpha) &= \{(q, \alpha)\} & \text{delete 1 X for every 1} \\
\delta(q, \varepsilon, Z_0) &= \{(q, \varepsilon)\} & \text{done}
\end{align*}
\]

And if you want to write a grammar, then convert using the algorithm, the grammar is
\[
S \rightarrow XC | AY \\
X \rightarrow aaXb | \varepsilon \\
Y \rightarrow bbYc | \varepsilon \\
A \rightarrow Aa | \varepsilon \\
C \rightarrow Cc | \varepsilon
\]

The idea is to use the non determinism of the PDA. You add one or two symbols each time you read a 0, and delete one each time you see a 1.

c) $0^n1^m \mid n \leq m \leq 2n$

c) $0^n1^m \mid n \leq m \leq 2n$

And if you want to write a grammar, then convert using the algorithm, the grammar is
\[
S \rightarrow 0S11 | 0S1 | \varepsilon
\]