2. Constructing four grammers

(a) $S \to 0S1 \mid 1S0 \mid 0S0 \mid 1S1 \mid 1$

(b) $S \to 1S1 \mid C$
$C \to 0C1 \mid \varepsilon$

C generates all binary strings that have same number of 0s and 1s with all 0s before 1s, it enables part "0^n" be always after part "1^m" which can only be added in rules of variable S.

(c) $S \to 0S \mid S0 \mid C$
$C \to D \mid D11 \mid 1D \mid 0110$
$D \to E \mid E1 \mid 1E \mid 010$
$E \to E0 \mid 00$

E generates binary strings contains at least 2, at most infinite zeros, D generates binary strings with at least 2 0s and at most one 1 if "1" is at the begining or end, and a special case "010".

C generates binary strings with at least 2 0s and at most 2 1s if "1" is at begining or the end, and a special case "0110"

This grammer works because it ensures at least 2 0s and at most 2 1s in special and general cases, then add whatever 0s you like to the begining and end of string.

(d) $S \to 1S1 \mid S11 \mid 11S \mid 10C \mid C$
$C \to 0C1 \mid C11 \mid 00C \mid \varepsilon$

C generates binary strings with n 0s and p 1s where $n \equiv p \pmod 2$, this grammar works because in rules of S and C, p mod 2 is always same with m+n mod 2 by making mod 2 of addition to p and m+n to be same. I can add 2 to m+n without addition to p, add 2 to p without addition to m+n, or add 1 to m+n and add 1 to p at same time.