1. For each of the following languages, give a CFG that generates it.

(a) All binary strings such that reading left to right the 1's never exceed the 0's. For example, 01001 is in the language, since each of 0, 01, 010, 0100, and 01001 has at least as many 0's as 1's.

\[ S \rightarrow 0S | TS | \epsilon \]
\[ T \rightarrow 0T1T | \epsilon \]

(b) All palindromes from alphabet \{x, y\} with an even number of x's

\[ P \rightarrow xPx | yPy | y | \epsilon \]

2. Consider the following CFG with start variable S:

\[ S \rightarrow A00A \]
\[ A \rightarrow 0A | 1A | 0 | 1 \]

(a) Give one string of length 4 in the language.

\[ 0000 \]

(b) Give one string of length 4 NOT in the language.

\[ 1111 \]

(c) Does this grammar generate a regular language?

\[ Yes \]

(d) Is this grammar ambiguous?

\[ Yes \]
3. Consider following PDA.

(a) Give two strings of length 4 accepted by the PDA.
(b) Give two strings of length 4 NOT accepted by the PDA.
(c) Describe in succinct-ish English the language of this PDA. Be precise.

All binary strings whose first and last bit are different.