1. (8 points) Describe a Turing Machine to accept $L$, the set of all binary strings with a length divisible by 3. For example $110010 \in L$ and $\varepsilon \in L$ though $0101 \notin L$.

2. (8 points) Are the following questions decidable?
   a. INPUT: Alphabet $\Sigma$, string $w \in \Sigma^*$, CFGs $G_1, G_2$ over $\Sigma$.
      QUESTION 1: Does $w \in L(G_1) \cap L(G_2)$?
   b. INPUT: Alphabet $\Sigma$, CFG $G$ over $\Sigma$.
      QUESTION 1: Does $\varepsilon \in L(G)$?

3. (10 points) Prove that language $L$ is recursive if and only if we can enumerate it in order of nondecreasing length. That is, $L$ is recursive if and only if we can list $z_0, z_1, \ldots$ such that $(x \in L) \iff (\exists i) x = z_i$ and $(i > j) \rightarrow (|z_i| \geq |z_j|)$. 

1. We let $b$ denote the blank tape symbol.
\[
\delta(s, \text{blank}) = (q_0, \text{blank}, R)
\]
\[
\delta(q_0, 0) = \delta(q_0, 1) = (q_1, 0, R)
\]
\[
\delta(q_1, 0) = \delta(q_1, 1) = (q_2, 0, R)
\]
\[
\delta(q_2, 0) = \delta(q_2, 1) = (q_0, 0, R)
\]
\[
\delta(q_0, b) = \delta(t, b) = (t, b, R)
\]
\[
\delta(q_1, b) = \delta(q_2, b) = \delta(r, b) = (r, b, R)
\]

2. The questions are each decidable.
   a. We can convert $G_1$ and $G_2$ to equivalent grammars in Chomsky Normal form. We can then try all derivations of length $2|w|-1$ is each grammar to test if $w$ belongs to the language that each grammar generates.
   b. $\epsilon \in L(G)$ if and only is $S \in \text{NULLABLE}$, the set of nonterminals which can be erased.

3. If we can list the members of $L$ in order of nondecreasing lengths, then to test whether a string $x$ belongs to $L$ we just examine the list until either we find a member with length greater than $|x|$ (in which case we reject $x$) or we encounter $x$ (in which case we accept it).

   If $L$ is recursive, then there is a total Turing machine $M$ to accept it. The following machine lists $L$.
   \[
   \text{for } n \leftarrow 0
   \quad \text{for each } z \in \Sigma^*, |z| = n
   \quad \text{if } z \in L(M) \text{ then } "\text{output } z" \\
   \quad n \leftarrow n + 1
   \]