Collaboration Policy: For this homework, each student should work independently and write up their own solutions and submit them.

Read the course policies before starting the homework.

• Homework 3 tests your familiarity with Turing Machines.
• Each student must submit individual solutions for these homework problems.
• Please carefully read the course policies on the course web site. If you have any questions, please ask in lecture, or by email. In particular:
  ○ Please submit all your homework questions in PDF form, preferably typed up in LaTeX.
  ○ Submit separately stapled solutions, one for each numbered problem, with your name and ID clearly printed on each page.
  ○ You may use any source at your disposal: paper, electronic, human, or other, but you must write your solutions in your own words, and you must cite every source that you use (except for official course materials). Please see the academic integrity policy for more details.
  ○ No late homework will be accepted for any reason.
  ○ Answering “I don’t know” to any (non-extra-credit) problem or subproblem, on any homework or exam, is worth 25% partial credit.
  ○ Unless explicitly stated otherwise, every homework problem requires a proof.
1 Required problems

1. (30 pts.)
Construct a Turing Machine (formal description) with input alphabet $\Sigma = \{a, b\}$ to insert a 
# symbol between each of the input symbols.

2. (30 pts.)
We have informally covered the notion of shifting tape contents by one symbol for Turing-
machine programs. We need to move the contents of each of the cells to the right of the current 
head position one cell right, and then find our way back to the current head position.

   (a) Give a high level description of how to perform this operation.

   **Hint**: Leave a special symbol to mark the position to which the head must return.

(b) Design a Turing machine that shifts the entire input string one cell to the right. In this 
part, you are supposed to give a formal description of the machine. Precisely, you will 
design a Turing machine $M$ such that, given an input string $w \in \{0, 1\}^*$, $M$s accepting 
configuration will be $\epsilon_{\text{acc}}#w$ (recall that “configuration” refers to the collection of symbol 
read, machine state, and string written on the tape).

3. (40 pts.)
Design a Turing Machine implementation (pseudo-code only, no formal diagram needed) for 
the following operations.

   (a) Doubles a number given on its input tape in unary representation, i.e. given a number $1^n$ Turing machine halts after writing $1^{2n}$ on its tape.

(b) Halves a number given on its input tape in unary representation, i.e. given a number $1^n$ Turing machine halts after writing $1^{\lfloor \frac{n}{2} \rfloor}$ on its tape.

(c) Doubles a number given on its input tape in binary representation. For example, given 
a number 01010 Turing machine halts after writing 010100 on its tape.

(d) Halves a number given on its input tape in binary representation. For example, given a 
number 0101 Turing machine halts after writing 010 on its tape.

(e) Accept the language $\#x\#y\#z\#$, where $x, y, z \in \{0, 1\}^*$, and $x + y = z$, i.e. binary string 
$x$ is equal to the binary addition of $y$ and $z$. 

2