Ella, Josh, Takuto

Worksheet #7
Turing Machines

1. Does a TM always terminate its execution at some point?

No!! The execution only stops when it hits an accept or reject state.

2. Give a formal description of a TM that uses a binary alphabet and accepts all even length input strings.
$\{ww : w \in \{0, 1\}^*\}$

q_0: start state
q_x: an odd number of symbols have been counted
q_y: an even number of symbols have been counted
q_{accept}: accept state
q_{reject}: reject state

$q_0 0 \rightarrow q_x 0R$
$q_0 1 \rightarrow q_x 1R$
$q_0 \square \rightarrow q_{accept}$

$q_x 0 \rightarrow q_y 0R$
$q_x 1 \rightarrow q_y 1R$
$q_x \square \rightarrow q_{reject}$

$q_y 0 \rightarrow q_x 0R$
$q_y 1 \rightarrow q_x 1R$
$q_y \square \rightarrow q_{accept}$

3. Write a trace of the TM execution for Question #2 for the input string 0110 by showing the TM's state, tape contents, and matched instruction for each step.

STATE	TAPE CONTENTS	INSTRUCTIONS
q_0	**0**110	$q_0 0 \rightarrow q_x 0R$
q_x	0**1**10	$q_x 1 \rightarrow q_y 1R$
q_y	01**1**0	$q_y 1 \rightarrow q_x 1R$
q_x	011**0**	$q_x 0 \rightarrow q_y 0R$
q_y	0110□	$q_y \square \rightarrow q_{accept}$
q_{accept}		

4. At the implementation level, define a TM that accepts the language, L = {x#y : x != y}. This computation checks that two strings are not equal.

1. Start by looking at the first non-crossed off symbol, record it, then mark it off
    a. If first non-crossed off symbol is #, move to 'confirmation state,' where you move to the leftmost non-crossed off symbol after the #. If it is anything but □, move to accept state. If it is □, move to reject state.
    b. If first non-crossed off symbol is not #, record it and move to step 2
2. Check if the first unmarked symbol to the right of the # matches.
    a. If yes, do not change state, cross off symbol, then move to leftmost unmarked symbol. Repeat from step 1.
    b. If no, move to accept state
    c. If □, move to the leftmost non-crossed off symbol in x. If that is #, move to reject state. If it is any other symbol, move to accept state.

5. At the implementation level, define a TM that accepts the language, L = {x#y : x is a binary string and y is the same string with all the 0's removed}.

1. Start by looking at the first symbol non-crossed off symbol (at the beginning of the execution this is just the first symbol in the string).
    a. If it is a zero, cross it off and move one to the right.
    b. If it is a one, cross it off, change state to the 'searching for a corresponding one state'
    c. If it is #, change to 'confirmation state,' where you check that everything following the # before

the □ has been crossed off. If you find a character that is not crossed off before □, move to reject state. If not, move to accept state.

2. If in 'searching for a corresponding one state', move right until pointing at leftmost symbol after the # (that is also not crossed off)
   a. If the pointer is looking at a zero, move to reject state
   b. If pointer is looking at a one, cross off the one and move into q_L
   c. If pointer is looking at a x(symbol crossed off), move to reject state
3. q_L: move one to the left, changing nothing, until hitting x. Then move one to the right, repeat from step one.