Due 12/15 (last day of final exams week)

Construct a Turing machine that subtracts one ternary integer (base 3) from another. An input will be of the form X#Y, where X and Y are elements of $\{0, 1, 2\}^+$. In particular, $X = x_n x_{n-1} \ldots x_1 x_0$, and $Y = y_m y_{m-1} \ldots y_1 y_0$, with each x_i, y_i in {0, 1, 2}, $X = (x_n \times 3^n) + (x_{n-1} \times 3^{n-1}) + \ldots + (x_1 \times 3^1) + (x_0 \times 3^0)$, and $Y = (y_m \times 3^m) + (y_{m-1} \times 3^{m-1}) + \ldots + (y_1 \times 3^1) + (y_0 \times 3^0)$. Your Turing machine must be a single tape, one way infinite, deterministic Turing machine. For this program you can use the left, right, and stay directives. When the Turing machine completes the tape should contain Z, where Z = X - Y. You do not need to delete X#Y from the tape, you can simply position the Turing machine read/write head at the beginning of Z. For this assignment you will probably want to use blocks (kind of like Turing machine subroutines in JFLAP) to build your Turing machine. Your Turing machine cannot make use of the blank spaces to the left of the input string and leading zeros are to be removed from Z (again, you can simply position the read/write head at the appropriate place). JFLAP has some unhappiness with filenames containing special characters and I don't know all the symbols that cause problems (I stick with alphanumeric, and have had problems with $ and # - JFLAP throws an error when trying to save the program after inserting the block).

There us no guarantee that X > Y, so you will need to put a negative sign in front of some of the results. It took me 3 – 4 hours to implement the program and about an hour to test it. I will test your program with 20 test strings. Each test string has $50 \le |X|, |Y| \le 100$. Each test string takes about 1 second to run with my program, and half of the string have $X \ge Y$ and half of the strings have $X < Y$. For my Turing machine I have $\sum$ = { 0, 1, 2, 3 #} and Γ = { 0, 1, 2, #, $, a, b, c, ' ', <, >}.

My Turing machine uses blocks for the following actions.

- Insert terminator at the left end of the tape (7 states)
  - I put two $s at the left end of the input as a way to determine if X and Y have been swapped (that is, to determine if a minus sign needs to by placed at the left end of the result)
- Rewind the tape (2 states)
  - I only use this a couple of times, since I have the same functionality hard coded in multiple places
  - Using this more would most likely reduce the number of states in a couple of the blocks
- Set X larger (3 states)
  - Updates a marker that keeps track of wether X is larger than Y
  - When comparing X and Y, I use a “>” at the end of the input to denote $X \ge Y$
- Set Y larger (3 states)
  - Updates a marker that keeps track of wether X is larger than Y
  - When comparing X and Y, I use a “<” at the end of the input to denote $X < Y$
- Correct order (23 states, 5 blocks – some may be duplicates)
  - Compare X and Y to determine which is larger, if $X < Y$, then they are swapped and a minus sign is place in front of the result
  - Based on the symbol at the end of the input (“<” or “>”), we know if $X \ge Y$ or $X < Y$
  - If “<” is at the end of the input, then X and Y are swapped (actually X is copied to the right of Y)
  - As X and Y are compared, from right to left, the symbol (“<” or “>”) at the end of the string is updated each time the current xi and yi don't match
  - I don't bother forcing X and Y to have the same length when performing this comparison
- Append 0/1/2 (3 states)
  - Used when swapping X and Y to put the current symbol at the tail end of the input

Due 12/15 (last day of final exams week)

- Insert 0/1/2 (6 states)
  - Used to insert the current value of xi – yi at the left end of the output
- Perform subtraction (34 states, 5 blocks – some may be duplicates)
  - Does the actual subtraction
  - Processes X and Y from right to left
- Main program (1 state, 7 blocks – some may be duplicates)
  - Initializes the data and calls the various blocks

E-mail the JFLAP file to me (david.garrison@binghamton.edu) by 11:59:59.999pm on the date due. The filename must be your last name followed by "_p7.jff" (as an example, my filename would be "garrison_p7.jff"). The subject of your e-mail is to be "CS 373 program 7".