CSC Fall 2023
HW #1
25 points/10 extra credit points

Submit a .zip archive that can contain these files:

- A plain text .txt file that affirms the honor code. For individuals: `On my honor as an SFSU student, I, <name>, have neither given or received inappropriate help with this homework assignment.` For groups: `On our honor as SFSU students, we, <names>, have neither given or received inappropriate help with this homework assignment. All group members participated in this work, and all concur with the submission.`
- A .zip archive containing your .jff file(s) for problem 1.
- A .py Python source code file with your answer for problem 2

1. (12.5 points/10 extra credit points for a minimal correct solution. Include the JFLAP file in your .zip submission). The input to TM M is $n_1\$n_2$, where n_1 and n_2 are strings of *1's* and *0's*. If $n_1 = n_2$, the output is $n_1=n_2$; when n_1 and n_2 are interpreted as the binary numbers m_1 and m_2 respectively, leading zeroes ignored, the output is $n_1<n_2$ if $m_1 < m_2$, and $n_1>n_2$ if $m_1 > m_2$. A correct implementation of M will have the transducer results below for the inputs in test_cases.txt, included with the test materials.

Input	Output	Result
0101$101	0101=101	Accept
0101$10	0101>10	Accept
0101$100	0101>100	Accept
0101$111	0101<111	Accept
0101$1010	0101<1010	Accept
101$101	101=101	Accept
101$10	101>10	Accept
101$100	101>100	Accept
101$111	101<111	Accept
101$1011111	101<1011111	Accept
11$011	11=011	Accept
11$01	11>01	Accept
11$010	11>010	Accept
11$00111	11<00111	Accept
110$011	110>011	Accept
11$11	11=11	Accept
11$1	11>1	Accept
11$10	11>10	Accept
11$0111	11<0111	Accept
110$011	110>011	Accept

HW1.jff, also included with the test materials, encodes a partial implementation of M. It handles $n_1 = n_2$ correctly, as well as the cases where $m_1 < m_2$, or $m_1 > m_2$. But when n_1 or n_2 starts with a `0` it simply gives up and changes the `$` to a `?`.

Enhance H1.jff to implement M. To run the test cases in transducer mode, select Multiple Run (Transducer) from the Input menu, and then select Load Inputs from the buttons at the bottom of the right panel. Note that transducer output extends right from the current symbol to the first blank (□).

2. **(12.5 points. Include a modified version of yesToSlow.py in your .zip submission)**

slowerThanInputLength(P) = 'yes' iff ∀I(P executes more than |I| Python statements before terminating). For example:

```
def ƒ(inString):
    result = 'z'
    for ch in 'abc':
       result += ch

    return 'done'
```

ƒ is not a positive instance of slowerThanInputLength because it always executes 6 statements (counting the for loop header as a single statement), but it is possible that len(inString) > 5.

Prove that slowerThanInputLenght is not decideable by a reduction from yesOnString, following the instructions in the yesToSlow.py template file.