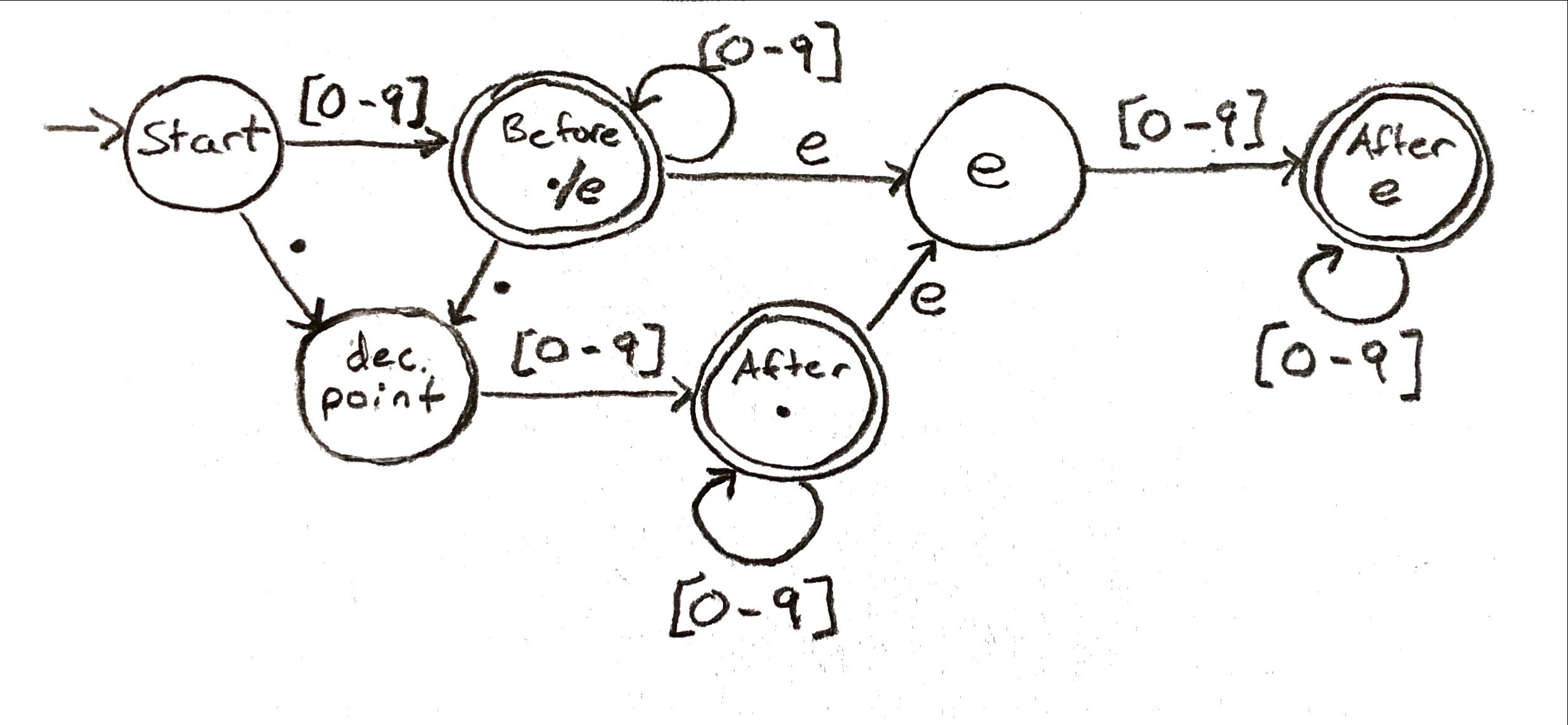
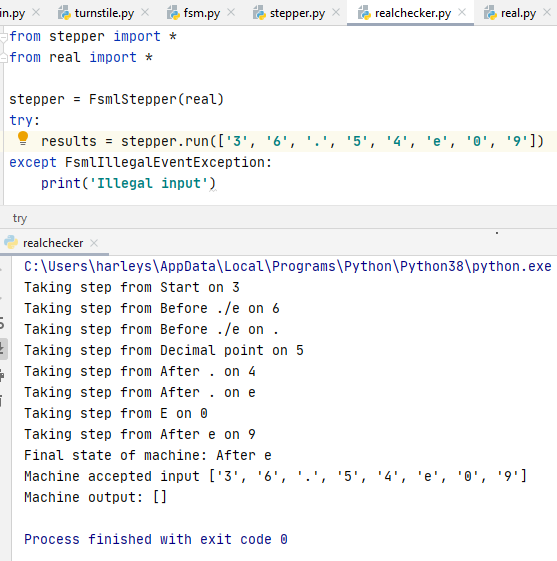
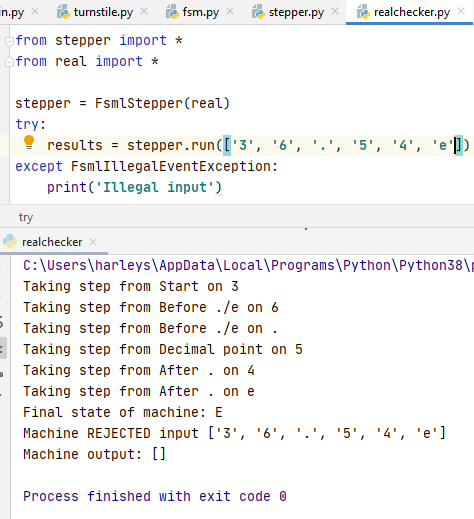
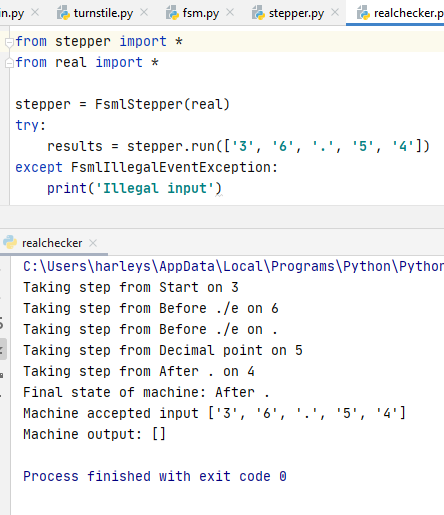
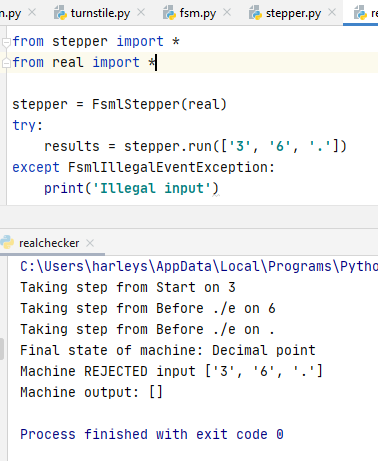
Stuart Harley

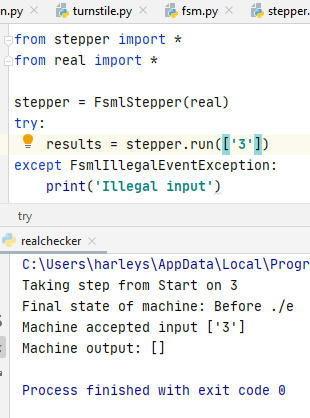
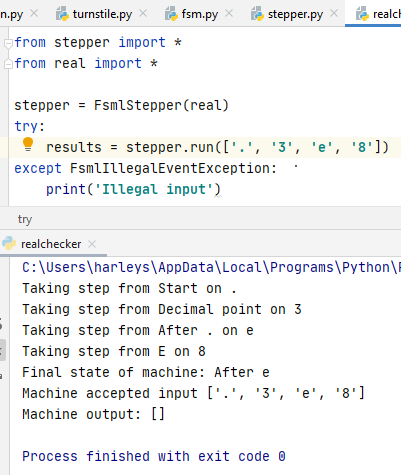
Assignment 1

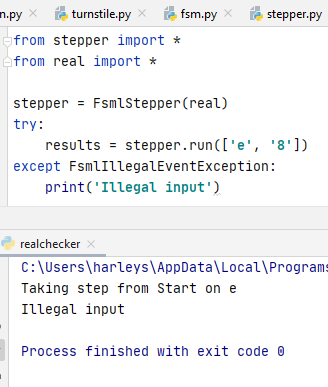
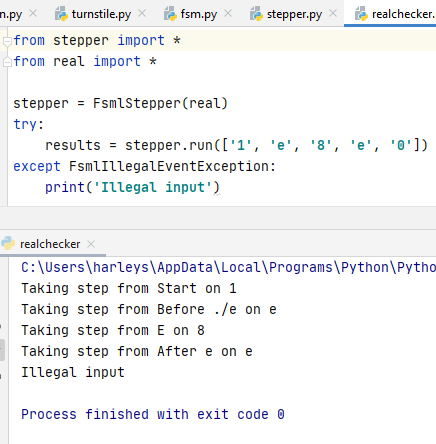
FSM for a machine accepting unsigned real numbers:

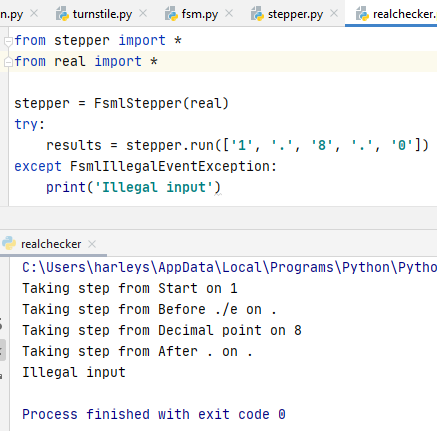
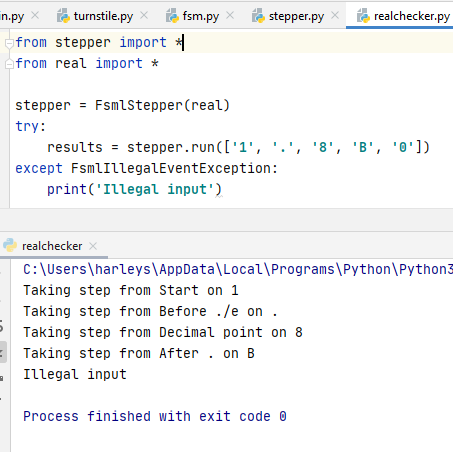
Test Output:

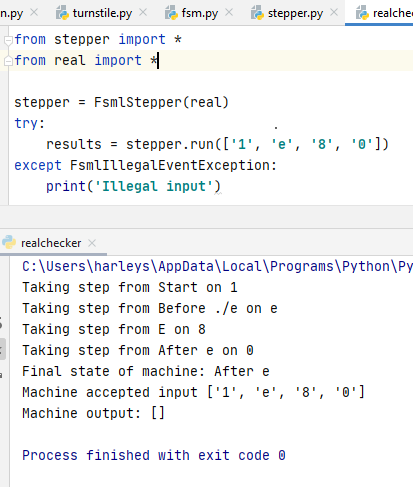
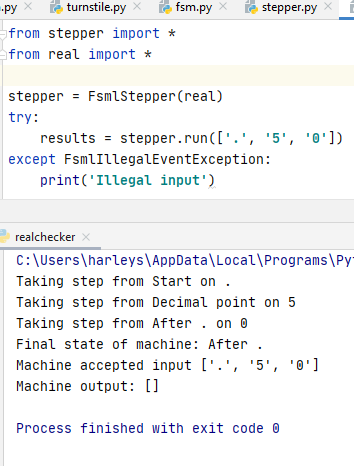
 

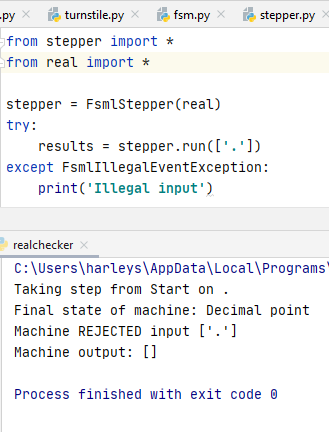
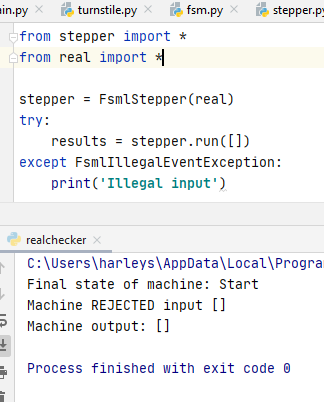
 

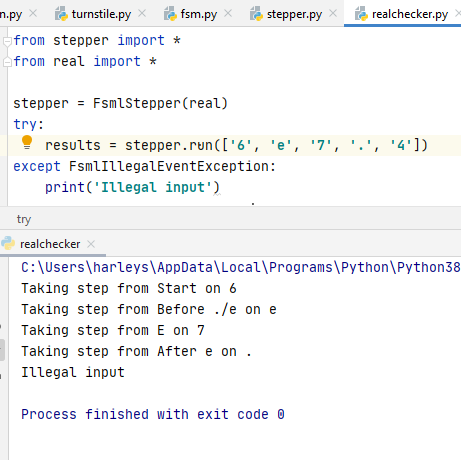
 



Analysis of Tests:

These inputs were accepted by the machine: 36.54e09, 36.54, 3, .3e8, 1e80, .50

These inputs were rejected by the machine because they ended in a non-final state: 36.54e, 36., . (just a decimal point), and the empty string.

These inputs threw a FsmlIllegalEventException, and so it was caught and “Illegal Input” was outputted: e8, 1e8e0, 1.8.0, 1.8B0, 6e7.4

I believe that these test cases capture every possible path through the real FSM, and each test behaves as it should. Valid inputs that ended in a final state were accepted. Valid inputs that ended in a non-final state were rejected by the machine. And non-valid inputs threw an exception which was caught and dealt with accordingly.