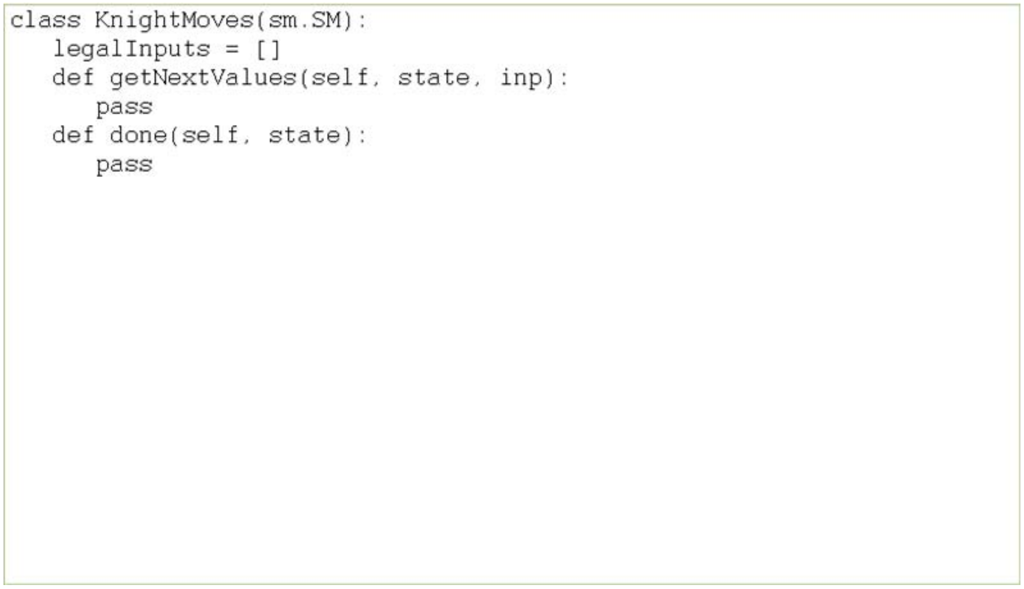
Problem Wk.13.3.6: Knight paths on a chessboard [Optional]

We want to write a search to find a path of knight's moves from a given initial square to a given goal square on a chessboard.

- The states will each be a pair of integers (*i*,*j*), designating a square on the chessboard. Use tuples to represent the pairs.
- According to the rules of chess, a knight on a chessboard can move two squares vertically and one square horizontally, or two squares horizontally and one square vertically. In general, there are 8 such moves possible. You can assign the names of the actions arbitrarily; the action names will not affect the correctness of your answer. One useful choice of actions are tuples that specify the offsets of the move.
- Make sure to check that the knight remains on the board at each step. A standard chessboard is 8x8, so the position indices must be between 0 and 7 (inclusive). Any illegal move should not change the state.

In the space below enter a state machine definition of this domain. The state machine should have `legalInputs` attribute, a `getNextValues` method and a `done` method that will terminate the search at the state `(7, 7)` (the upper right hand corner of the chessboard). Don't worry about the `startState`.



```
class KnightMoves(sm.SM):
   legalInputs = []
   def getNextValues(self, state, inp):
      pass
   def done(self, state):
      pass
```

6.01SC Introduction to Electrical Engineering and Computer Science
Spring 2011