### **SIENA COLLEGE**

**30th Annual** High School Programming Contest

##### **March 24, 2017**

###### Gold Problem #6: Snakes and Ladders

Background Information: According to the classic board game, Snakes and Ladders, a player, starting at space zero, is attempting to reach square 100, on a board with snakes, that transport a player from square N to square M < N, and ladders, that transport a player from square N to square M > N. A player moves a random number of spaces by rolling a standard six-sided die and then transports if they land at the bottom of a ladder or top of a snake. Given the destination square, the locations of snakes and the location of ladders, can you determine the least number of rolls needed to reach the destination square?

* The Destination Square is the last square on the board.
* A player rolls a die, moves, and then determines if a snake/ladder is involved. The totality of these actions counts as one roll of the die.
* Ladders and Snakes will never intersect at their endpoints.

###### Programming Problem:

Input:  A positive integer P on between 100 and 10,000 inclusive followed by a non-negative S (number of snakes) and a non-negative L (number of ladders) L We are then followed by S + L lines of pairs, where the first S lines are the source square and destination square of the snakes. The next L lines are the source square and destination square of the ladders.

Output: The minimal number of rolls needed to reach the destination. If a solution is not possible, print NOT POSSIBLE.



###### Example 1: Input:  100 10 9

16 6

49 11

62 19

87 24

47 26

56 53

64 60

93 73

95 75

98 78

1 38

4 14

9 31

21 42

28 84

36 44

51 67

71 91

80 100

###### Output:  7

V1