Function convert(location):

column = location[0]

If column == 'A':

Return 0

Else if column == 'B':

Return 1

Else if column == 'C':

Return 2

Else if column == 'D':

Return 3

Else if column == 'E':

Return 4

Else if column == 'F':

Return 5

Else if column == 'G':

Return 6

Else if column == 'H':

Return 7

Else if column == 'I':

Return 8

Else:

Return 9

Function re\_convert(column):

If column == 0:

Return 'A'

Else if column == 1:

Return 'B'

Else if column == 2:

Return 'C'

Else if column == 3:

Return 'D'

Else if column == 4:

Return 'E'

Else if column == 5:

Return 'F'

Else if column == 6:

Return 'G'

Else if column == 7:

Return 'H'

Else if column == 8:

Return 'I'

Function check\_row(board, row, num):

For each column in board:

If board[row - 1][column] == num:

Return False

Return True

Function check\_column(board, column, num):

For each row in board:

If board[row][column] == num:

Return False

Return True

Function check\_square(board, row, column, num):

square\_row = row // 3 \* 3

square\_col = column // 3 \* 3

For i from square\_row to square\_row + 2:

For j from square\_col to square\_col + 2:

If board[i][j] == num:

Return False

Return True

Function check\_location1(board, row, column\_a):

row -= 1

column = convert(column\_a)

If !(row < 0 OR row >= 9 OR column < 0 OR column > 8):

Print "ERROR: Square {column\_a}{row} is invalid"

Return False

Function check\_location2(board, row, column):

If board[row - 1][column] != ' ':

Print "ERROR: Square {re\_convert(column)}{row} is filled"

Return False

Function valid\_input(num):

If num is not in ['1', '2', '3', '4', '5', '6', '7', '8', '9', 'S']:

Print "ERROR: The value {num} is invalid"

Return False

Function helper(board, row, column):

valid\_numbers = []

For num from '1' to '9':

If check\_row(board, row, num) AND check\_column(board, column, num) AND check\_square(board, row, column, num):

valid\_numbers.append(num)

print valid\_numbers

