**Knight Tours Analysis:**

Board is an array of 64 ints : BDIM 8, int board [BDIM \* BDIM] = board[8 \* 8]

**Main:**

Prompt the user for a seed number: 123

Srand (seed)

**Gen\_sol:**

Initialize the board cells to -1

**Initial Board status of Knights Tour**

-1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 -1 -1 -1 -1 -1

-1 -1 -1 -1 -1 -1 -1 -1

Increment count of gen\_sol calls

Assign startx, starty with rand () % BDIM the starting position of the knight

Mark the board cell as 1 board [y \* BDIM + x] = 1

Loop BDIM \* BDIM times

Call make\_move with x and y position, and check return code

**Make\_move:**

Get start value with rand () % BDIM

Loop BDIM times

Compute the subscript with (start + count) % BDIM

Get new values of x and y, newx = oldx + increment[subscrip], newy = oldy + increment[subscript]

Get neighbor count with newx and newy

**get\_neighbor\_count:**

Loop BDIM times

Increment count if the neighbor of newx and newy is free with the increments of x and y

**Is\_freee:**

x and y are positive, x and y are less than BDIM board cell is marked -1