Offline 02: Hill Climbing and Simulated Annealing

Solve the 4-queen problem using steepest ascent hill climbing algorithm and simulated annealing algorithm. [12+18]

i. The implementation is partially done for you in here. You can follow this code or do it by your own

ii. In the code, we maintain two data structures one is **queen_loc** (in which cell queen is located) and **chess_board**(N*N). **addQueen()** method add queens randomly. **get_conflict(Q)** finds the number of conflict with the queen **Q** using the method **conflict(). calc_cost(state)** calculated the total cost of the given **state** and return cost with the queen who has maximum conflicts. **get_neighbor(row,col)** tries to find the neighbor of the given cell. **Assume that** a queen can **move** within the **same column**.

iii. Now your task to solve 4-queen problem.

iv. You can set your initial state as below so that easily trace code is correct or not

	Q		Q
Q		Q	