Subject: Artificial Intelligence   **SAP ID: 60004220253 – Devansh Mehta**

Experiment No. 07

Aim: To implement a game using AI.

Theory:

N-queens problem:
The N-Queens problem is a classic combinatorial problem in chessboard arrangement. The objective is to place N chess queens on an N×N chessboard in such a way that no two queens threaten each other. This means that no two queens should share the same row, column, or diagonal.

For example, in the 8-Queens problem, you need to place 8 queens on an 8x8 chessboard in such a way that no two queens are in the same row, column, or diagonal.

Solving the N-Queens problem using Prolog involves representing the problem in terms of facts and rules and then utilizing Prolog's backtracking mechanism to explore possible solutions. A common approach is to represent the state of the board as a list of rows, where each row contains the column number of the queen in that row.

Code:

Knowledge Base:

```
nqueens(N, Queens) :-
    numlist(1, N, Rows),
    permutation(Rows, Queens),
    safe(Queens).

safe([]).
safe([Q|Queens]) :-
    safe(Queens, Q, 1),
    safe(Queens).

safe([], _, _).
safe([Q|Queens], Q0, D0) :-

    Q0 =\= Q,
    abs(Q0 - Q) =\= D0,

    D1 is D0 + 1,
    safe(Queens, Q0, D1).
```

Output:

Subject: Artificial Intelligence   **SAP ID: 60004220253 – Devansh Mehta**

nqueens(4, Queens).

Queens = [2, 4, 1, 3]
Queens = [3, 1, 4, 2]
false

nqueens(8, Queens).

Queens = [1, 5, 8, 6, 3, 7, 2, 4]
Queens = [1, 6, 8, 3, 7, 4, 2, 5]
Queens = [1, 7, 4, 6, 8, 2, 5, 3]

Next | 10 | 100 | 1,000 | Stop

nqueens(12, Queens).

Queens = [1, 3, 5, 8, 10, 12, 6, 11, 2, 7, 9, 4]

12.076 seconds cpu time

Queens = [1, 3, 5, 10, 8, 11, 2, 12, 6, 9, 7, 4]

0.605 seconds cpu time

Queens = [1, 3, 5, 10, 8, 11, 2, 12, 7, 9, 4, 6]

```
?-   nqueens(12, Queens).
```