Termklork - 5

- Problem Statement
  Design an algorithm for so To solve eight queens problem and develop a prolog program for the same

- Theory:
  The eight queens problem is prolla of placing 8 queens on an 8x8 Chessboard such that none of attacd are another more generaly then. n queens problem places n queen an nxn Chess board

  The pseudocode uses a backtracking algorithm to find a solution on Chess board is such a way that no two queens threaden each other

  The algorithm starts by playing a queen on first column then it procedes next colourn and planes a queen it first of row that columns.

  If algorithm reaches 8th column and all queens are placed in selt positioning it the board & between tree this is sate forchie 'checks' if it is sate to place a queen an a certain row or column by checking it in same row diagonal or anti-diagonal

  It is worth to notice that this it just a high level pseudocode and it light need to be adopted dependency on specific implementation and language.

Program:

```
#include<stdio.h>
#include<math.h>

void queen(int row, int p);
int Chess[8], count;

int main(){
    int p=8;
    queen(1,p);
    return 0;
}
void print(int p)
{
    int i,j;
    Printf("This is solution no. %d: \n\n", ++(count));
    for(i=1; i<=p; ++i)
        Printf("\t%d",i);
    for(i=1; i<=p; ++i){
        Printf("\n\n%d",i);
        for(j=1; j<=p; ++j){
            if(chess[i]==j)
                Printf("\t1");
            else
                Printf("\t-");
        }
    }
    Printf("There are total 92 solutions for 8-queens
        Problem;");
}
```

```
int place(int row, int column)
    {
      int i;
      for (i=1; i<=row, int column)
      {
        if(chess[i]==column)
          return 0;
        else
        if(abs(chess[i]-column)==abs(i-row))
         return 0;
      }
      return 1;
}
void queen(int row, int p)
  {
    int column;
    for(column=1; column<=p; ++column)
    {
      if(place(row, column))
        if(row==p)
          print(p);
        else
          queen(row+1, p);
      }
  }
}
```

Output :-

Soln 1 :

```
- - - q - - - -
- - - - - q - -
- - q - - - - -
- - - - - - q -
- q - - - - - -
- - - - q - - -
q - - - - - - -
- - - - - q - -
```

Conclusion :- In this term work we learnt about the 8 queen problem and how to solve it with prolog.