

## Project 4: N-Queens

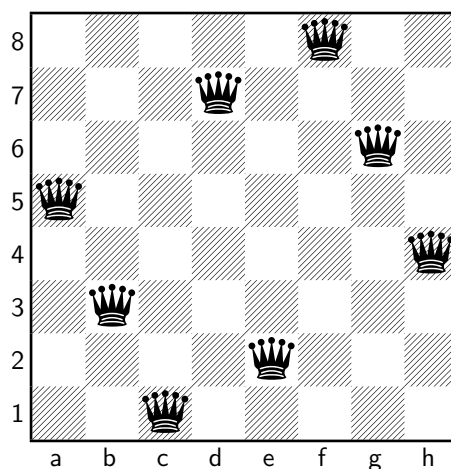
### INSTRUCTIONS

1. This project must be done and submitted individually.
2. Submit one ZIP file, called `<student number>.zip` to the folder "Project 4 Submissions" in Luminus by **Friday 12 April midnight**.
3. The ZIP file contains four directories: `Prolog`, `File`, `DIMACS3`, `DIMACS8`, and `DIMACSmax`. the `Prolog` directory contains a file `nqueens.pl`, a Prolog program with your answer for question 1. the `File` directory contains a file `nqueensDIMACS.pl`, a Prolog program with your answer for question 2. The `DIMACS3`, `DIMACS8`, and `DIMACSmax` directories contain the DIMACS input and output files for questions 3 and 4.
4. After the deadline and until Friday 19 April at midnight, you can submit to the folder "Project 4 Late Submissions" in Luminus (penalties apply).
5. Note that you may be asked to demonstrate your code on your machine.

The N-Queens problem is the problem of placing  $n$  queens on a  $n \times n$  chessboard in such a way that they do not threaten each other, namely, that no two queens are on the same row, column, or diagonal. The following Prolog code illustrates how to open or create a file as well as how to write to and read from it.

```
createOpenFile(N):-
    term_string(N, SN), string_concat("nqueen.", SN, File),
    open(File, write, St),
    writeln(St, "Hello world"),
    close(St).

readFile:-
    open("nqueen.8", read, St),
    read_string(St, end_of_line, X, L),
    write(L).
close(St).
```



We consider a numbering of the chessboard squares from left to right and from bottom to top from 1 to  $n^2$ . On the  $8 \times 8$  chessboard represented on the figure above the eight queens are on positions on squares 3, 13, 18, 32, 33, 47, 52 and 62, respectively.

**Question 1** [5 marks]

Write a Prolog (do not use Constraint Logic Programming features of ECLiPSe) procedure `nqueens(+Nat, ?Solution)` with `Nat` being the number of queens,  $n$ , to be placed on a commensurate chess board, that succeeds when `Solution` is a list, in increasing order, of the  $n$  squares on which one of the  $n$  queen can be safely placed.

For example, `nqueens(8, [3, 13, 18, 32, 33, 47, 52, 62])` succeeds.

For example, `nqueens(8, S)` returns all the solutions through backtracking.

Explain your code in the comments. Indicate in the comments the references to the sources (books, articles, Web documents) that you have consulted. Submit the Prolog file `nqueens.pl` with the program (procedure and ancillary procedures).

**Question 2** [5 marks]

Write a Prolog procedure `nqueensFile(+Nat, +File)` with `Nat` being the number of queens,  $n$ , to be placed on a commensurate chess board, that creates a file `File` in DIMACS format for MiniSat that contains a SAT representation of the corresponding N-Queens problem. Do not implement any other constraint other than the one stated in the problem statement above (e.g. do not exploit symmetry or other possible features of the problem).

Explain your code in the comments. Indicate in the comments the references to the sources (books, articles, Web documents) that you have consulted. Submit the Prolog file `nqueensDIMACS.pl` with the program (procedure and ancillary procedures).

**Question 3** [5 marks]

Submit the DIMACS input files that you created and used to find all the solutions to the N-Queens problem for  $n = 3$  and for  $n = 8$  together with the corresponding DIMACS output files with the solutions.

**Question 4** [5 marks]

Submit the DIMACS file that your program created to find one solution to the N-Queens problem for  $n = M$ , where  $M$  is the maximum number for which you were able to find one solution. Give the solution that you found in the form of the DIMACS output file. The marks for this questions are assigned competitively.

– END OF PAPER –