

LAB 2

Write Prolog predicates to solve the following:

1) Compute the greatest common divisor of two integers – gcd/3;

```
gcd(0,X,X):-!.
gcd(X,0,X):-!.
gcd(A,B,C):-A=<B, D is B-A, gcd(D,A,C).
% ... the case for A>B – consider using !.
```

2) Split a list into 2 lists, according to a given value: the elements that are less than the value are placed in the first list and the elements greater or equal than the value are placed in the second list – split/4;

```
split([],H,[],[]).
split([H1|T],H,[H1|A],B):-H1=<H, split(T,H,A,B).
% ... the case for H1>H – consider using !.
%?-split([3,2,6,1,5],4,L1,L2)
```

3) Insertion sort and quick sort – insertsort/2 and quicksort/2;

```
insertsort([],[]).
insertsort([X|T],S):-insertsort(T,ST),insert(X,ST,S).
```

```
%insert(Elem,OrderedList,OrderedList1) -- insert the element Elem on its right
%position in OrderedList and the result is OrderedList1
%Example. Elem=3, OrderedList=[1,2,6,9] => OrderedList1=[1,2,3,6,9]
```

```
quick([],[]).
quick([X],[X]).
quick([H|T],L):-split(T,H,A,B),quick(A,A1),quick(B,B1),...build the solution L
from the partial solutions A1 and B1
```

4) The 8 queens' problem – queen/1 - the solution may be represented as a list of lists. For example, `[[1,3],[2,5],[3,7],...,[8,1]]` represents the configuration where the 1st queen is on the 1st column and 3rd row, the 2nd queen on the 2nd column and the 5th row, ..., the 8th queen is on the 8th column and 1st row;

```
queen([]).
```

```
queen([[X,Y]|S):-queen(S),member(Y,[1,2,3,4,5,6,7,8]), not(attack([X,Y],S)).
```

```
attack([X,Y],S):-member([X1,Y1],S), (...write the conditions where the queen on
position [X,Y] attacks the queen on position [X1,Y1]).
```

```
%?-queen([[1,X1],[2,X2],[3,X3],[4,X4],[5,X5],[6,X6],[7,X7],[8,X8]]).
```

5) Given the following knowledge base:

There are 5 houses, each of a different color (red, white, blue, yellow, green).

In each house, there is only one person, of a different nationality (British, Swedish, Danish, Norwegian, German).

Each person of each house drinks a certain drink (milk, beer, tea, water, coffee), smokes a certain brand of cigarettes (PallMall, Winfield, Marlboro, Dunhill, Rothmans) and owns a certain animal (bird, dog, horse, cat, fish).

Each of the 5 people drinks a different drink, smokes a different brand of cigarettes and owns a different kind of animal.

The British man lives in the red house.

The Norwegian lives next to the blue house.

The green house is to the (immediate) left of the white house.

The owner of the green house drinks coffee.

The owner of the house in the middle drinks milk.

The owner of the yellow house smokes Dunhill.

The Norwegian lives in the first house.

The Swedish man has a dog.

The person who smokes Pall Mall has a bird.

The Marlboro smoker lives next to the one with a cat.

The Winfield smoker drinks beer.

The person who owns a horse lives next to the one who smokes Dunhill.

The German smokes Rothmans.

The Marlboro smoker has a neighbor who drinks water.

write a Prolog program that answers the question “Who has the fish?”.

Suggestions for implementation:

```
einstein(Sol) :- Sol=[[1,N1,C1,P1,D1,S1],
```

```
                [2,N2,C2,P2,D2,S2],
```

```
                [3,N3,C3,P3,D3,S3],
```

```
                [4,N4,C4,P4,D4,S4],
```

```
                [5,N5,C5,P5,D5,S5]],
```

```
member([_,englishman,red,_,_,_],Sol), % The Britishman lives in the red house
```

```
member([WH,_,white,_,_,_],Sol),      % The green house is on the immediate left of the white house
```

```
member([GH,_,green,_,_,_],Sol),
```

```
GH == WH - 1,
```

```
...
```

```
%implement here the rest of the rules
```

OR

einstein(Sol) :- Sol=[[N1,C1,P1,D1,S1],

[N2,C2,P2,D2,S2],

[N3,C3,P3,D3,S3],

[N4,C4,P4,D4,S4],

[N5,C5,P5,D5,S5]], implement here all the rules making use of the predicate member

and define additional predicates to check different positions of elements in a list (e.g. next, left, first).