Exercise Sheet: Prolog

1.) Find the YAP webpage (http://www.ncc.up.pt/~vsc/Yap/) and download the latest version of YAP. Install YAP (either the installer.exe or the binaries.zip). You can also download it from www.karwath.org/teaching/winter2007/ba/files .

2.) Open a command prompt (Start-Run-cmd). In the terminal enter the following command:

```
set PATH=c://Yap/bin;%PATH%
```

3.) Download the file underground.pl from www.karwath.org/teaching/winter2007/ba/files/ and save it to your preferred directory (in the following this directory will be called YOURDIR).

4.) In your terminal cd to YOURDIR, start yap (yap.exe + ENTER), and load the file underground.pl ([underground]. + ENTER).

5.) In a text editor load the file underground.pl to inspect the program.

6.) In your terminal where YAP is running, experiment with different queries:

```
?- nearby(X,Y).
?- nearby(B,A).
?- nearby(freiburg,X).
…
```

7.) The trace command invokes the debugger (trace. + ENTER). Invoke it and try your queries again ('h': gives you a list of commands for the debugger)

8.) Download the file monkey.pl from the same website

9.) Experiment with some queries in Yap. Use the debugger to trace the execusion.

10.) Try to write your own new predicate called sumOfOdd/2, summing up all odd number of a list of integer numbers. The query:

```
?- sumOfOdd([1,2,3,4,5,6],O).
```

should give:

```
O = 9
```

as a result

11.) Using the predicate append/3 (below), try to write a predicate reverse/2, which reverses a list.

```
append([],L,L).
append([A|L],L1,[A|L2]) :- append(L,L1,L2).
```

You might want to program out different predicate, such as union/3 or intersection/3 which calculate the union (Vereinigung) or the intersection(Schnittmenge) of sets. Remember how to write the predicate member/2 ?

12.) Remember the eight queens problem. Define a predicate called solution/1 that is incrementally constructed. Consider two cases:

a. The list of queens is empty. There is no attack, so it's a solution.
b. The list of queens is non-empty. It then looks like [X/Y | Others]

In case 2 the following conditions need to be met:

- Others is a solution
- X and Y are integers between 1 and 8 (you might want to use the member predicate for that)
- The queen at X/Y does not attack any queen in (you will have to write a new noattach/2 predicate)

Here we use the following representation of the board:

```
[X1/Y1,X2/Y2,…,X8/Y8]
```

Where we can fix the X-coordinates:

```
[1/Y1,2/Y2,…,8/Y8].
```