David A. Elizondo – De Montfort University, Leicester, UK elizondo@dmu.ac.uk

Laboratory 1 – introduction to the SWI-Prolog Prolog interpreter exercises.

Through out this introductory course in Prolog you will be using the Prolog interpreter sw-prolog. You should try your best to understand how the interpreter works before moving on.

Introduction to SWI-Prolog

Learning outcomes

After this lab you will be able to use SWI-Prolog to do run simple Prolog programs, and you will be able to save your programs to files and to load them from file. You will know some of the simpler capabilities of the SWI-Prolog program. You will develop simple Prolog programs and run them using the SWI-Prolog interpreter.

Download the SWI prolog interpreter from :

http://www.swi-prolog.org/

and install it on your home computer

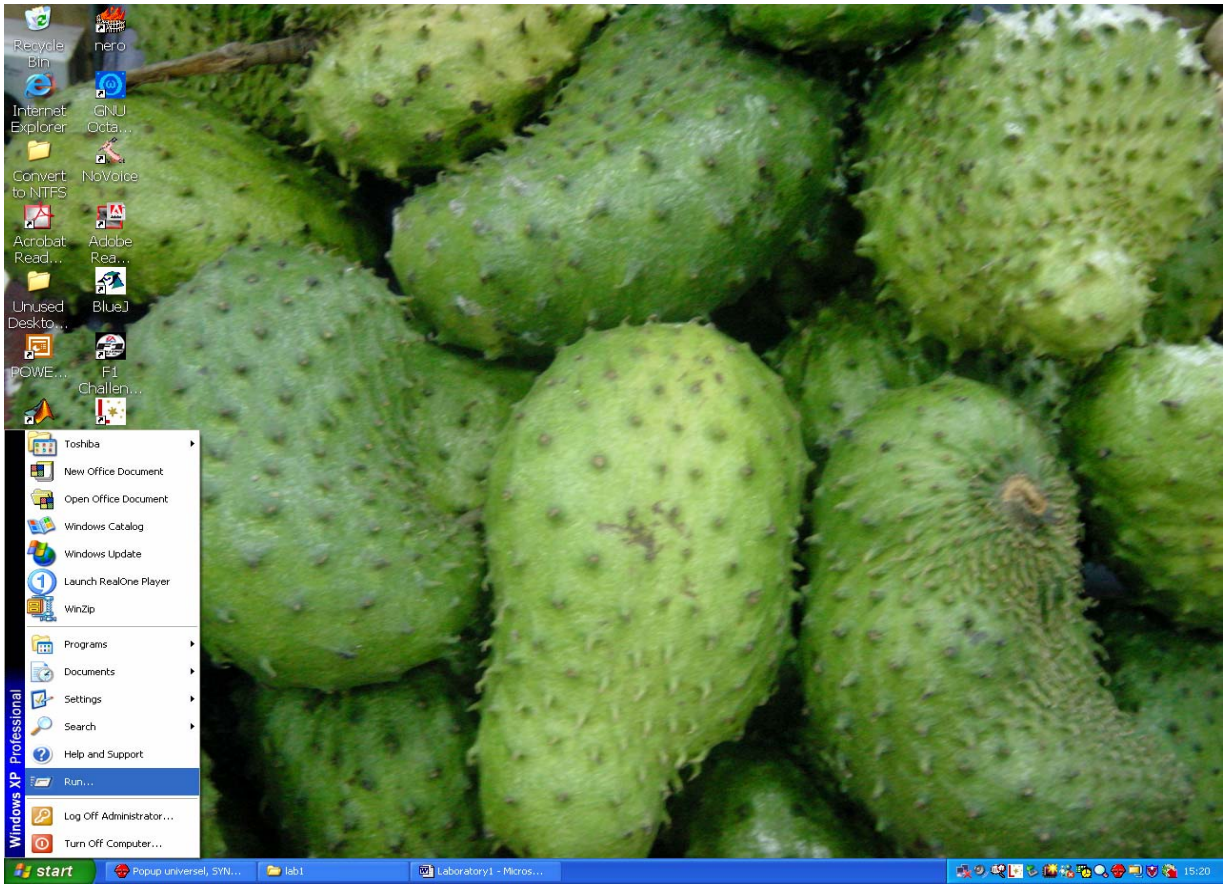
Create a directory called prolog within your home computer. Go to this directory and create a new subdirectory called lab1. Go to this directory.

Open up a session of notepad and enter the following Prolog program:

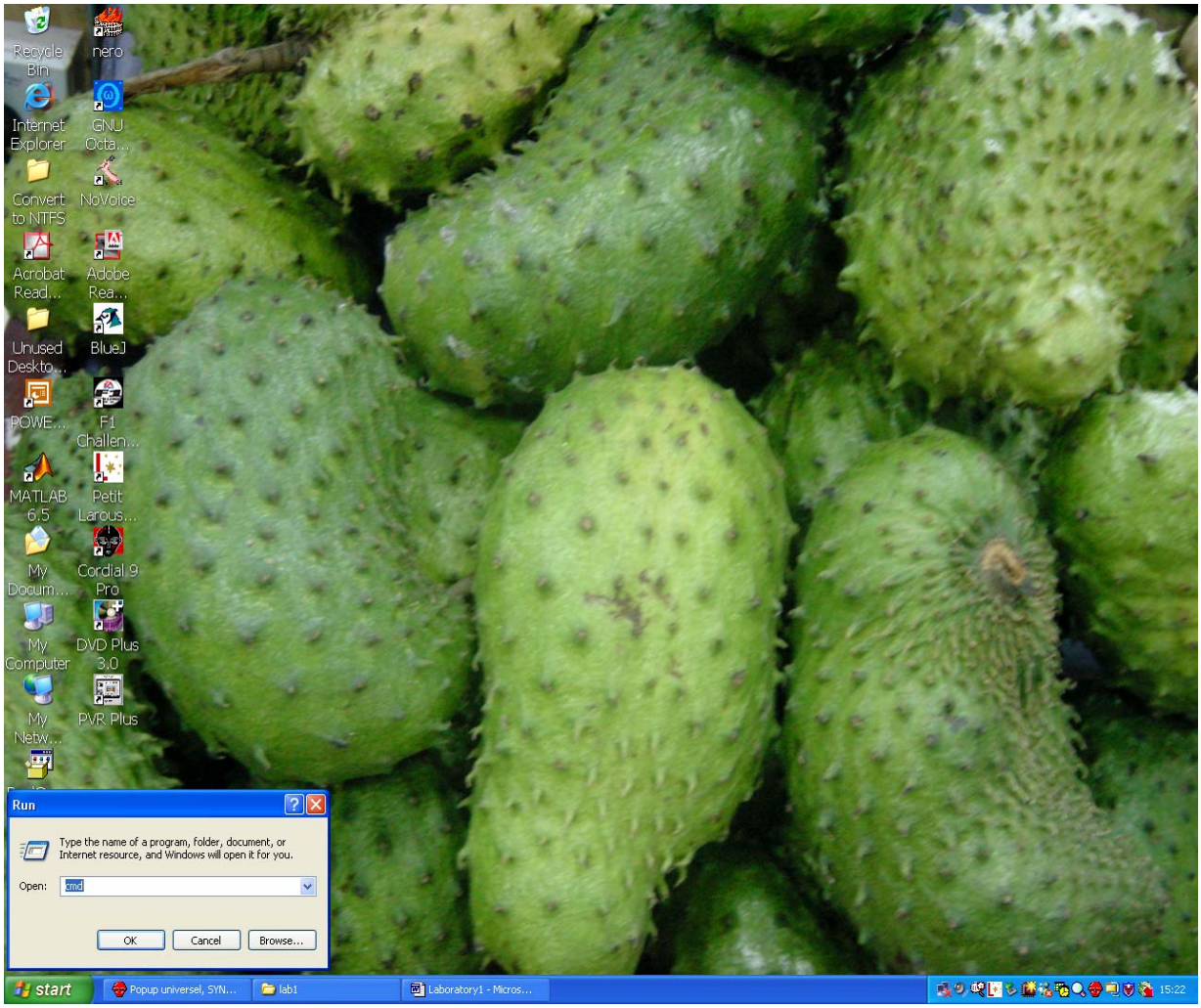
```
likes(tim,chips).
likes(tim,fish).
likes(tim,spinach).
likes(tim,apples).
likes(carla,oranges).
likes(carla,chicken).
likes(carla,chocolate).
likes(carla,ice_cream).
likes(tim,crepes).
```

Save the file in the current directory under the name likes.pl.

Make sure that you have named the file likes.pl.
Go to the star button on the left hand corner and select run

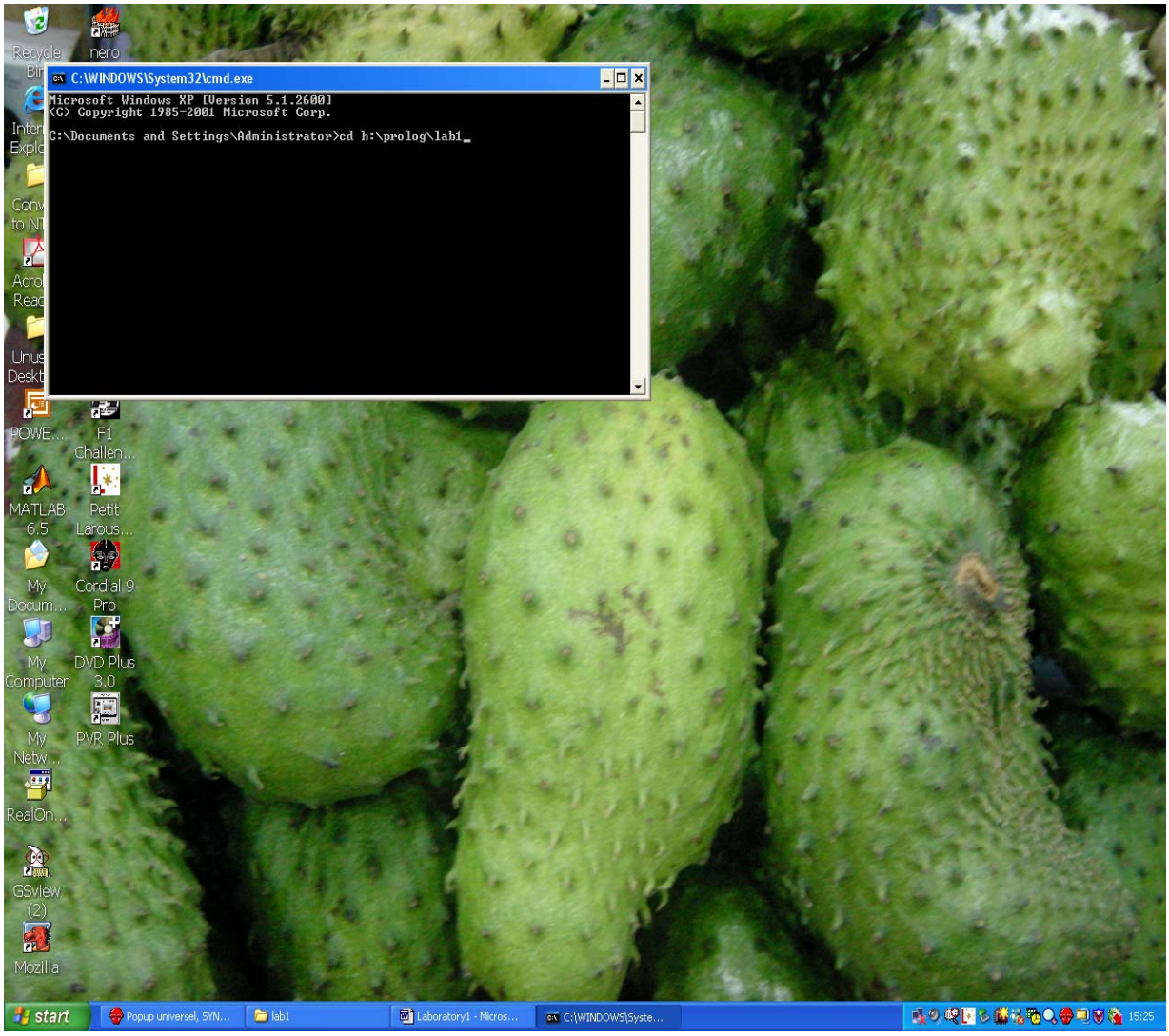


Type in cmd to open up an MS-DOS text window



Issue the MS-DOS command, for example:

cd h:\prolog\lab1\ and hit the <ENTER> key.



Now execute the command:

pl.exe

This should open up a Prolog session and look like this:

Welcome to SWI-Prolog (Multi-threaded, Version 5.2.13)
Copyright (c) 1990-2003 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?-

After executing the program, you can load the likes Prolog program by typing in the following command

?- [likes].

```
% likes compiled 0.00 sec, 0 bytes

Yes
?-
```

Prolog responds with Yes if everything went ok. Otherwise we will get the following error

```
?- [likes].
ERROR: source_sink `likes' does not exist
?-
```

NOTE: Do not forget to end the command with a **full stop**

After loading a program, one can ask Prolog queries about the program. The query below asks Prolog what food 'Tim' likes. The system responds X = value if it can prove the goal for a certain X. The user can type semi-colon (;) if (s)he wants another solution, or <ENTER> if (s)he is satisfied. after which Prolog will say Yes. If Prolog answers No, it indicates that it cannot find any (more) answers to the query. Finally, Prolog can answer using an error message to indicate the query or program contains an error.

Issue the following query and verify that you get a similar result:

```
?- likes(tim,X).

X = chips ;

X = fish ;

X = spinach ;

X = apples ;

X = crepes ;

No
?-
```

What query would you use to find out if Carla likes soup or not? Type in the query and verify that Carla does not like soup.

What query would you use to find out if anybody likes crepes? Type in the query and verify that Tim likes crepes.

Go back to the notepad editor and add the following fact to your program:

```
likes(carla,crepes).
```

Make sure that you save the file with these changes.

From within the Prolog window reload the file by typing:

?- [likes].

What query would you use to find out if Carla likes crepes or not? Type in the query and verify that Carla does like crepes.

Using a single query how would you verify that both Tim and Carla like crepes? Verify that this works.

To exit the program type in the command:

?- halt.

Now you can close the MS-DOS window and Notepad and finally log off the system.