

159.202 Assignment 2

Deadline:	Anytime before: Monday 10 Aug 2015, time due 11:45 am
Evaluation:	10 marks – which is 3% of your final grade
Late Submission:	25% per hour (or fraction of hour) it is late
Purpose:	Practice with Prolog lists and recursion.

You are expected to submit only a single file (*a2.pl*) containing all your answers.

Part A. [1 mark] Write a rule `output`, such that when the query `?- output` is used it will display **(all)** your ID(s) and name(s) on screen (like you did in assignment 1).

Part B [1 mark]. Define two Prolog operators: `=>>`, and `isSingleton` such that `=>>` is an infix binary operator that checks that one of its operands is double of the other operand. The `isSingleton` operator is a unary prefix operator that checks that its list operand has exactly one element.

Here are some examples of using the operators:

```
?- 40 =>> 20.
true.
?- (-8) =>> (-4).
true.
?- 0 =>> 0.
true.
?- 7 =>> 3.
false.
?- isSingleton [].
false.
?- isSingleton [-8].
true.
```

Part C. [5 marks]

a) Define a predicate `totalInList(L,N)` which, given a list of integers `L`, returns the sum `N` of all the elements of `L`.

b) Define a predicate `count(Word,List,N)` which is true when `N` is the number of times that the word `Word` occurs in list `List`.

For example

```
count(my, [the, friend, of, my, enemy, is, not, my,
           enemy,nor,my,friend], N)
```

should be true for `N = 3`.

c) Define a predicate `adjAve(L,Ave)`, that is true when `Ave` contains the list of the averages of each consecutive two numbers in list `L`. If the list `L` contains an odd number of elements the last one is ignored.

For example

```
?-adjAve([4,6,7,20,9],A).
```

should give the answer: `A=[5.0,13.5]`.

d) Write a Prolog program to perform translation between English and French words for the first 10 positive integers.

Write

- i) a fact `word/2` which succeeds when its first argument -- an English word and its second argument -- a **French** word, represent the same number between 1 and 10. Use the following data¹, for creating the knowledgebase of the word facts:

1	2	3	4	5	6	7	8	9	10
One	Two	Three	Four	Five	Six	Seven	Eight	Nine	Ten
Un	Deux	Trois	Quatre	Cinq	Six	Sept	Huit	Neuf	Dix

- ii) a predicate `translate(EnglishNr, FrenchNr)` which translates a list of English words representing numbers (only numbers from one to ten) to the corresponding list of French number words. For example:

`?-translate([one,nine,ten],X).`

should produce:

`X = [un,neuf,dix].`

Part D [3 marks]

A Latin square² is a $n \times n$ square filled with n different Latin letters, each occurring exactly once in each row and exactly once in each column.

Here is an example of a Latin square for $n=4$:

a b c d
d a b c
c d a b
b c d a

Write a Prolog program to generate and display on screen all Latin square of size 4. Your program should contain the predicate `generateSq` defined as:

`generateSq(SQ):-fillSq(SQ),testSq(SQ),displaySq(SQ).`

You have to write the three predicates:

- `fillSq` (non-recursive) to fill the 16 cells in the square with the values a, b, c and d;
- `testSq` to check that the square is a Latin square and
- `displaySq` (recursive) to “pretty-print” the square.

Here is one possible running of the program:

¹ From <http://www.maori.org.nz/kotereo/default.asp?pid=sp149&parent=115>

² See http://en.wikipedia.org/wiki/Latin_square

```

1 ?- generateSq(A).
      -----
      | a | b | c | d |
      -----
      | d | a | b | c |
      -----
      | c | d | a | b |
      -----
      | b | c | d | a |
      -----
A = [[a, b, c, d], [d, a, b, c], [c, d, a, b], [b, c, d, a]] ;
      -----
      | a | b | c | d |
      -----
      | d | a | b | c |
      -----
      | b | c | d | a |
      -----
      | c | d | a | b |
      -----
A = [[a, b, c, d], [d, a, b, c], [b, c, d, a], [c, d, a, b]] ;
      -----
      | a | b | c | d |
      -----
      | b | d | a | c |
      -----
      | c | a | d | b |
      -----
      | d | c | b | a |
      -----
A = [[a, b, c, d], [b, d, a, c], [c, a, d, b], [d, c, b, a]] ■

```

If you have any questions about this assignment, please ask the lecturer before its due time!

Submit your file for marking using the 159.202 Stream submission page.

Important:

1. You can define and use extra predicates in order to solve a problem. Use only the material presented in lectures, Notes (stream) or Labs; marks will be deducted if the solution uses material not covered in lectures or in the above mentioned places.
2. The assignment can be done individually or in pairs (pair programming). All assignments authored by 3 or more students will get 0 marks.
3. **Submit only one solution file per team. Different files having the same authors will get 0 marks.**
4. Please note that if we cannot run your a2.pl you will get 0 marks.
5. As sample solutions will be presented in next day lecture no extension will be possible.