

Time limit: 120 minutes. Total score is 20.

Make sure you have SWI prolog installed!

Get the zip file from your Courseville assignment page and unzip/extract them into your computer.

Each question's file must be done in its corresponding folder.

1. (3 marks) A familyQuiz.pl program, with all facts for parent and age, is provided in folder Question1:

Create file familyFunc.pl (Don't forget to add :-reconsult('familyQuiz.pl'). in its first line) in folder Question1. Write the following prolog predicate.

youngestGrandparent(X)

This predicate is true when X is the youngest grandparent.

Example runs of familyFunc.pl are as follows:

```
?- youngestGrandparent(X).
X = b2 ;
X = b2 ;
false.
```

2. (6 marks) Create a file 'hanoi.pl' in folder Question2. Then write the following function:

- a) (5 marks) **hanoi(N,A,B,C)**. This predicate prints out all the moves - for moving **N** disks from peg **A** to peg **B**, with peg **C** as a spare peg. The moves must follow the rule for tower of Hanoi (you are allowed to search for rules of the moves and how to write the code in C++, Java and Python). All parameters are considered to be input.

For checking, the following queries give the following results:

<pre>?- hanoi(1,a,b,c). Move disk 1 from peg a to peg b. true . ?- hanoi(3,a,b,c). Move disk 1 from peg a to peg b. Move disk 2 from peg a to peg c. Move disk 1 from peg b to peg c. Move disk 3 from peg a to peg b. Move disk 1 from peg c to peg a. Move disk 2 from peg c to peg b. Move disk 1 from peg a to peg b. true .</pre>	<pre>?- hanoi(4,a,b,c). Move disk 1 from peg a to peg c. Move disk 2 from peg a to peg b. Move disk 1 from peg c to peg b. Move disk 3 from peg a to peg c. Move disk 1 from peg b to peg a. Move disk 2 from peg b to peg c. Move disk 1 from peg a to peg c. Move disk 4 from peg a to peg b. Move disk 1 from peg c to peg b. Move disk 2 from peg c to peg a. Move disk 1 from peg b to peg a. Move disk 3 from peg c to peg b. Move disk 1 from peg a to peg c. Move disk 2 from peg a to peg b. Move disk 1 from peg c to peg b. true .</pre>
---	---

b) (1 mark) **hanoisave(N,A,B,C)**. This works just like **hanoi(N,A,B,C)**, but moves are not printed on screen. Instead, they are written to a file 'result.pl'.

3. (6 marks) Create a file **removeN.pl** in folder Question3.
Write predicate **removeN(N,List1,Remain,Removed)**.

This predicate returns true when:

- **List1** is a list (considered as an input).
- **N** is the number of data (from the front of the list) that you want to remove from **List1** (considered as an input).
- **Remain** is List1, after removing the first N data (considered as an output).
- **Removed** is the list of **N** data removed from **List1**, in the order that they are removed (considered as an output).

For checking, the following queries give the following results:

<p>?- removeN(-1,[a,b,c,d,e,f],X,Y). X = [a, b, c, d, e, f], Y = [].</p> <p>?- removeN(0,[a,b,c,d,e,f],X,Y). X = [a, b, c, d, e, f], Y = [].</p> <p>?- removeN(1,[a,b,c,d,e,f],X,Y). X = [b, c, d, e, f], Y = [a].</p>	<p>?- removeN(3,[a,b,c,d,e,f],X,Y). X = [d, e, f], Y = [a, b, c].</p> <p>?- removeN(5,[a,b,c,d,e,f],X,Y). X = [f], Y = [a, b, c, d, e].</p> <p>?- removeN(7,[a,b,c,d,e,f],X,Y). X = [], Y = [a, b, c, d, e, f].</p>
--	---

4. (5 marks) Create a file **maxCommon.pl** in folder Question4.
Write predicate **maxCommon(List1,List2,M)**. This predicate is true when **List1** and **List2** are input lists of integer and **M** is the largest value that is in both **List1** and **List2**. **M** is considered to be an output.

For checking, the following queries give the following results:

<p>?- maxCommon([], [3,4,5], M). <u>false.</u></p> <p>?- maxCommon([1,2], [], M). <u>false.</u></p> <p>?- maxCommon([1,8,7], [3,4,5], M). <u>false.</u></p>	<p>?- maxCommon([1,2,3], [3,4,5], M). <u>M = 3 ;</u> <u>false.</u></p> <p>?- maxCommon([10,1,8,7], [3,4,5,7,8,9], M). <u>M = 8 ;</u> <u>false.</u></p>
---	--

How to submit:

Zip all folders (Question1 to Question4) into {your_id}.zip (For example: 6131070221.zip) and attach it to assignment on Courseville.