

Recursive Procedures in Prolog

1.) A recursive definition of the factorial function is provided below:

$$n! = \begin{cases} 1 & \text{if } n = 0 \\ n \times (n - 1)! & \text{if } n > 0 \end{cases}$$

Write a recursive procedure named `factorial` which computes this function. The first parameter to `factorial` should be `n`, and the second parameter should be the result. For example, the following queries should succeed:

```
?- factorial(0, 1).  
?- factorial(3, 6).
```

```
factorial(0, 1).  
factorial(N, Result) :-  
    N > 0,  
    MinusOne is N - 1,  
    factorial(MinusOne, RestResult),  
    Result is N * RestResult.
```

2.) Write a procedure named `myBetween` (so named to avoid conflict with the predefined `between` procedure) which nondeterministically produces all the values within a range, inclusive. `myBetween` should take three parameters, namely:

1. The low end of the range
2. The high end of the range
3. A value within that range

Example queries and output involving `myBetween` is shown below:

```
?- myBetween(3, 5, X).  
X = 3 ;  
X = 4 ;  
X = 5.
```

```
?- myBetween(2, 2, X).  
X = 2.
```

```
?- myBetween(3, 2, X).  
false.
```

As a hint, the case where the low end is less than or equal to (`=<` in Prolog) the high end can serve as a base case.

```
myBetween(Low, High, Low) :-  
    Low =< High.  
myBetween(Low, High, Result) :-  
    Low < High,  
    NewLow is Low + 1,  
    myBetween(NewLow, High, Result).
```