CSI 2120

Lab 5: Prolog

Exercise 1. Flight Schedule

Given the following facts:

```
flight(montreal, chicoutimi, 15:30, 16:15).
flight(montreal, sherbrooke, 17:10, 17:50).
flight(montreal, sudbury, 16:40, 18:45).
flight(northbay, kenora, 13:10, 14:40).
flight(ottawa, montreal, 12:20, 13:10).
flight(ottawa, northbay, 11:25, 12:20).
flight(ottawa, thunderbay, 19:00, 20:30).
flight(ottawa, toronto, 10:30, 11:30).
flight(sherbrooke, baiecomeau, 18:40, 20:05).
flight(sudbury, kenora, 20:15, 21:55).
flight(thunderbay, kenora, 20:00, 21:55).
flight(toronto, london, 13:15, 14:05).
flight(toronto, montreal, 12:45, 14:40).
flight(windsor, toronto, 8:50, 10:10).
```

Which of the following predicates allows one to decide if one's arrival time at the airport is early enough to catch a certain flight. On-time arrival at the airport is at least 60 minutes prior to the corresponding departure time of the flight.

Rule Set 1:

```
on_time(H1 : _M1, D, A)  :-
        flight(D, A, H2 : _M2, _H3 : _M3),H2 - H1 > 1.
on_time(H1 : M1, D, A)  :-
        flight(D, A, H2 : M2, _H3 : _M3),
        H2 - H1 =:= 1, MM is 60 - M1,  MM + M2 >= 60.
```

Rule Set 2:

```
on_time(H1 : _M1, D, A)  :-
        flight(D, A, H2 : _M2, _H3 : _M3),H2 - H1 > 2.
on_time(H1 : M1, D, A)  :-
        flight(D, A, H2 : M2, _H3 : _M3),
        H2 - H1 =:= 1, MM is 60 - M1,  MM + M2 >= 60.
```

Rule set 3:

```
on_time(H1 : _M1, D, A)  :-
        flight(D, A, H2 : _M2, _H3 : _M3),H2 - H1 > 2.
on_time(H1 : M1, D, A)  :-
        flight(D, A, H2 : M2, _H3 : _M3),
        H2 - H1 =\= 1, MM is 60 - M1,  MM + M2 >= 60.
```

Rule Set 4:

```
on_time(H1 : _M1, D, A)  :-
        flight(D, A, H2 : _M2, _H3 : _M3),H2 - H1 > 1.
```

```
on_time(H1 : M1, D, A)  :-
        flight(D, A, H2 : M2, _H3 : _M3),
        H2 - H1 =:= 1, MM is 60 - M1,  MM - M2 < 60.
```

Exercise 2. Arrival Predicate

Write an arrival predicate which enables the following type of query with the database of Exercise 1:

```
?- arrival( flight(montreal,sherbrooke), X).
X=17:10
```

Exercise 3. Sum of Integers

Write a predicate that finds the sum of the first N numbers, i.e., it should enable queries of the form `sum_int(N,X).`

Exercise 4 and Quiz: Series

Please hand-in the answer to this question on Virtual Campus during your lab session but at the latest by Friday 6:00pm! Remember, your submission will only count if you have signed the lab attendance sheet.

We would like to define a predicate that calculates the cosine with a series approximation
$cos(z) = \sum_{k=0}^{\infty} \frac{(-1)^k z^{2k}}{(2k)!}$

Obviously, we can not calculate an infinte number of terms, instead our predicate will use an extra parameter N to determine the number of terms in the summation.

Fill the gaps fixing the following set of rules to correctly approximate the cosine value according to the above equation.

```
% Factorial from class
fact(0, 1).
fact(N, F) :- N > 0,
  N1 is N-1,
  fact(N1, F1),
  F is F1 * N.

% Calculate -1 ^ K
signCnt(0,1).
signCnt(K,S) :- K > 0,
  K1 is K - 1,
  signCnt(K1,S1),
  ____________________.

% Base case
cosN(_________,_________,_,0).

% Recursive case
cosN(K,N,X,Y) :- K < N,
             signCnt(K,S),
             K2 is 2 * K,
             fact(K2,F),
             Yk is (S * X**K2)/F,
             ________________________,

             ________________________,
```

_____.

```
cosN(N,X,Y) :- N>0, cosN(0,N,X,Y).
```

Example Output:

```
?- cosN(5,pi,Y).
Y = -0.9760222126236076 ;
false.


?- cosN(25,pi,Y).
Y = -1.0 ;
false.


?- cosN(3,pi/2,Y).
Y = 0.01996895776487828 ;
false.


?- cosN(10,pi/2,Y).
Y = -3.3306690738754696e-15 ;
false.
```