

HW3 LISP Report

執行環境：Ubuntu 16.04

Problem1:

The screenshot shows a SWI-Prolog IDE with three tabs: problem1.pl, problem2.pl, and problem3.pl. The code in problem1.pl is as follows:

```
1 isPrime(2) :- !.
2 isPrime(3) :- !.
3 isPrime(X) :-
4     X > 3,
5     X mod 2 =\= 0,
6     N_max is floor(sqrt(X)),
7     isPrime_(X,3,N_max).
8 isPrime_(X,N,N_max) :-
9     ( N > N_max
10    -> true
11    ; 0 =\= X mod N,
12      M is N + 2,
13      isPrime_(X,M,N_max)
14    ).
15 twoPrime(X) :-
16     Y is X - 2,
17     (
18         isPrime(Y)
19     -> output(2,Y),true
20     ; C is 2 + 1,
21       D is Y - 1,
22       twoPrime_(X,C,D)
23     ).
24 twoPrime_(X,Y,Z) :-
25     (
26         isPrime(Y), isPrime(Z)
27     -> output(Y,Z), true
28     ; C is Y + 1,
29       D is X - C,
30       twoPrime_(X,C,D)
31     ).
32 output(A,B) :-
33     write("Output: "), write(A), tab(1), write(B), nl.
34 main :-
35     write("Input: "), read(A),
36     A > 2, 0 =:= A mod 2,
37     twoPrime(A),halt.
38 :- initialization(main).
```

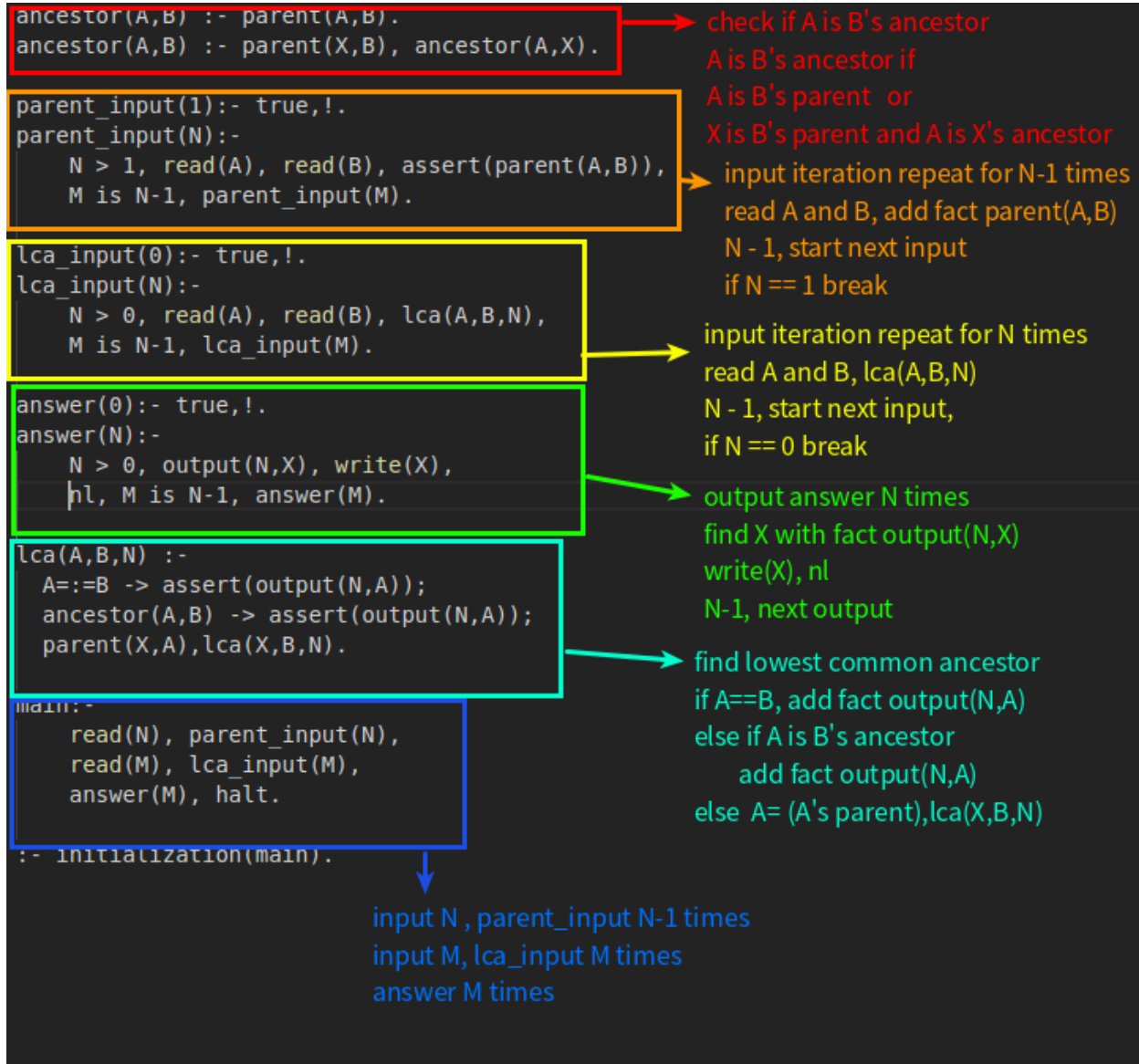
Annotations for the code:

- isPrime(2) :- !.** and **isPrime(3) :- !.**: Check if the X is prime number. 2 and 3 is definitely prime.
- isPrime(X) :-**
 - X > 3,** **X mod 2 =\= 0,** **N_max is floor(sqrt(X)),** **isPrime_(X,3,N_max).**: if X > 3, first check if X is even. if not, then divide X with odd numbers smaller than sqrt(X).
- isPrime_(X,N,N_max) :-**
 - (N > N_max -> true** **; 0 =\= X mod N,** **M is N + 2,** **isPrime_(X,M,N_max)** **).**: Find the two prime numbers that their sum is X. start with 2 and Y = X-2. if Y is prime, answer is 2 and Y. else C = 2+1, D = Y-1, check if C and D are prime numbers until C and D are both prime numbers.
- output(A,B) :-**
 - write("Output: "), write(A), tab(1), write(B), nl.**: output A and B. format: Output: A B.
- main :-**
 - write("Input: "), read(A),** **A > 2, 0 =:= A mod 2,** **twoPrime(A),halt.**: Input a number A. check A > 2 and A is even. find the two prime numbers.
- :- initialization(main).**: initialization(main).

執行方式&執行結果：

```
walltsai@walltsai-System-Product-Name:~/Programming_Language/PL_HW3$ swipl -q -s problem1.pl
Input: 100.
Output: 3 97
walltsai@walltsai-System-Product-Name:~/Programming_Language/PL_HW3$
```

Problem2:



執行方式&執行結果：

```
walltsai@walltsai-System-Product-Name:~/Programming_Language/PL_HW3$ swipl -q -s problem2.pl
|: 6.
|: 1. 2.
|: 2. 3.
|: 1. 4.
|: 4. 5.
|: 4. 6.
|: 3.
|: 3. 4.
|: 5. 6.
|: 1. 2.
1
4
1
walltsai@walltsai-System-Product-Name:~/Programming_Language/PL_HW3$
```

Problem3:

<pre>path(A,B) :- walk(A,B,[]). walk(A,B,V) :- edge(A,X), not(member(X,V)), (B = X ; walk(X,B,[A V])).</pre>	<p>→ check if there is a path between A B walk from A to B, V is a list of visited nodes if X has a edge with A, and X isn't in V either X==B, X is the destination or keep walking from X to B add A to V</p>
<pre>edge_input(0):- true,!. edge_input(N):- N > 0, read(A), read(B), assert(edge(A,B)), M is N-1, edge_input(M).</pre>	<p>→ input iteration repeat N times read A B, add fact edge(A,B) next loop</p>
<pre>con_input(0):- true,!. con_input(N):- N > 0, read(A), read(B), con(A,B,N), M is N-1, con_input(M).</pre>	<p>→ input iteration repeat N times read A B, check if A and B connected next loop</p>
<pre>answer(0):- true,!. answer(N):- N > 0, output(N,X), write(X), nl, M is N-1, answer(M).</pre>	<p>→ output iteration for N times find X with fact output(N,X) print X, next loop</p>
<pre>con(A,B,N) :- path(A,B) -> assert(output(N,"Yes")); assert(output(N,"No")).</pre>	<p>→ check A and B is connected or not if path(A,B) is true, add fact output(N,Yes) else add fact output(N,No)</p>
<pre>main:- read(N), read(E), edge_input(E), read(M), con_input(M), answer(M), halt. :- initialization(main).</pre>	<p>→ read N E, N nodes and E edges input the edges read M, check M queries output M answers</p>

執行方式&執行結果：

```
walltsai@walltsai-System-Product-Name:~/Programming_Language/PL_Hw3$ swipl -q -s problem3.pl
Warning: /home/walltsai/Programming_Language/PL_Hw3/problem3.pl:30:
Singleton variables: [N]
|: 6. 6.
|: 1. 2.
|: 2. 3.
|: 3. 1.
|: 4. 5.
|: 5. 6.
|: 6. 4.
|: 2.
|: 1. 3.
|: 1. 5.
Yes
No
walltsai@walltsai-System-Product-Name:~/Programming_Language/PL_Hw3$
```