SWISH File Edit Examples Help

Notebook

```
1 happy(X) :- man(X),rich(X),famous(X).
2 happy(X) :- woman(X),healthy(X).
3 happy(X) :- woman(X),wealthy(X).
4 happy(X) :- woman(X),wise(X).
5
6 man(jim).  man(bob).
7 healthy(jim). rich(jim). wise(jim).
8 woman(jane). wealthy(jane).
9 famous(bob).
```

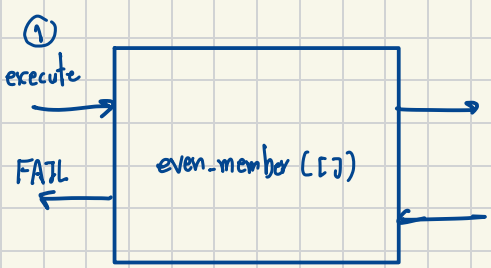
```
?- happy(jim).
false
```

```
?- happy(jane).
true
```

```
?- happy(bob).
false
```

even_member([]).

even_member([H|T]) :- even(H), even_member(T).

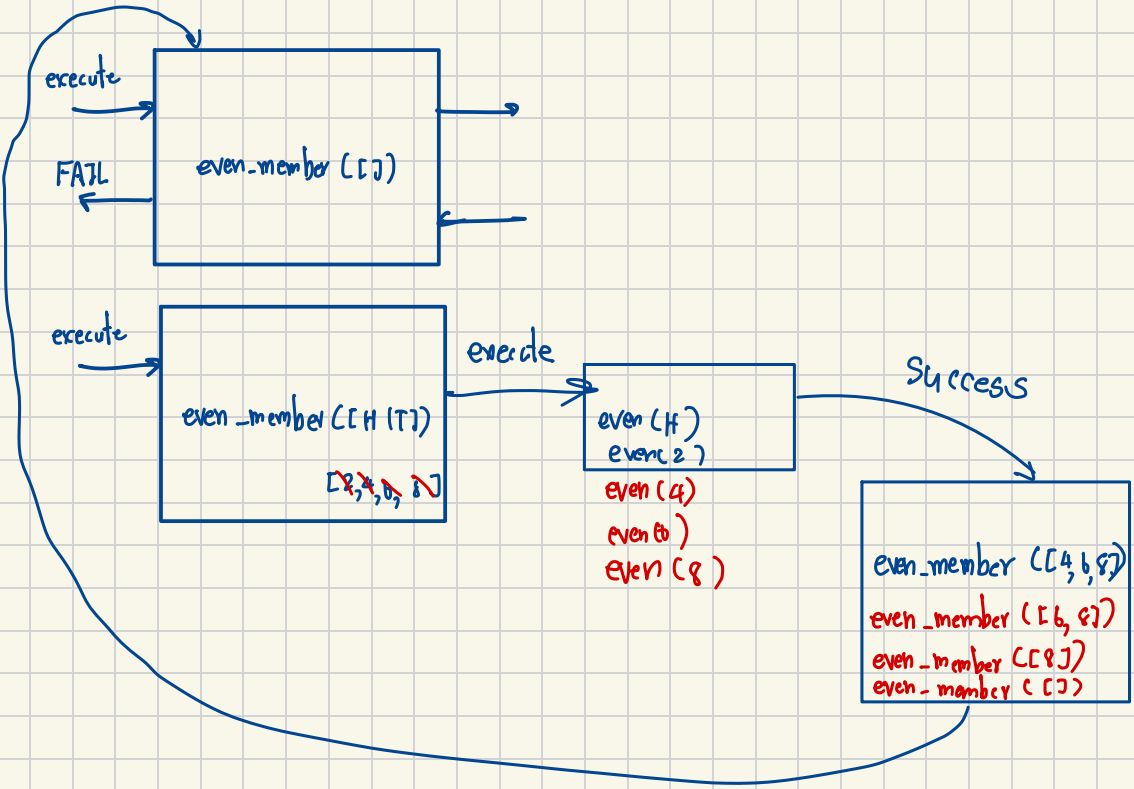


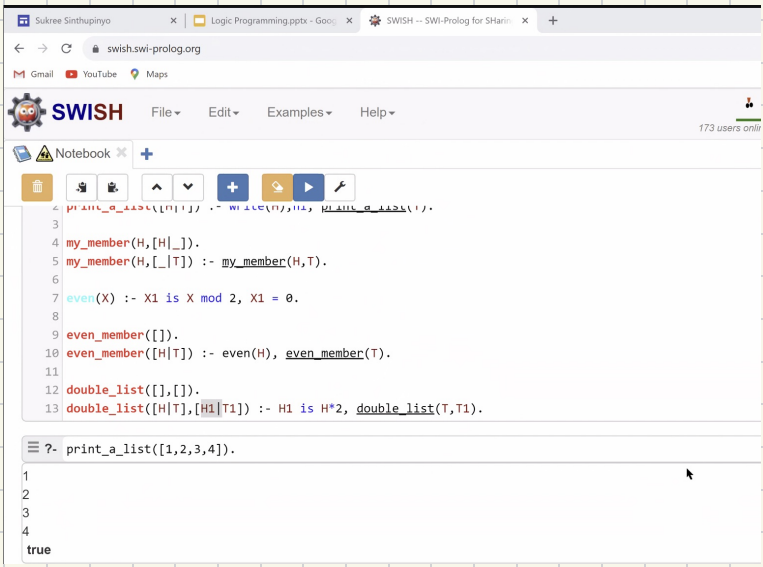
even_number([2,4,6,8])

execute(even_member([2,4,6,8]))

execute(even_member([])).

FAIL even_member([]).


execute
FAIL
even_member([])
execute
even_member([H|T])
[8,4,6,8]
execute
even(H)
even(2)
even(4)
even(6)
even(8)
Success
even_member([4,6,8])
even_member([6,8])
even_member([8])
even_member([])


Sukree Sinthupinyo
Logic Programming.pptx - Goo
SWISH -- SWI-Prolog for SHar
swish.swi-prolog.org
Gmail
YouTube
Maps
SWISH
File
Edit
Examples
Help
173 users onli
Notebook
my_member(H,[H|_]).
my_member(H,[_|T]) :- my_member(H,T).
even(X) :- X1 is X mod 2, X1 = 0.
even_member([]).
even_member([H|T]) :- even(H), even_member(T).
double_list([],[]).
double_list([H|T],[H1|T1]) :- H1 is H*2, double_list(T,T1).
?- print_a_list([1,2,3,4]).
1
2
3
4
true

Sukree Sinthupinyo | Logic Programming.pptx - Goo | SWISH -- SWI-Prolog for SHarin

swish.swi-prolog.org

Gmail YouTube Maps

SWISH File Edit Examples Help

168 users onli

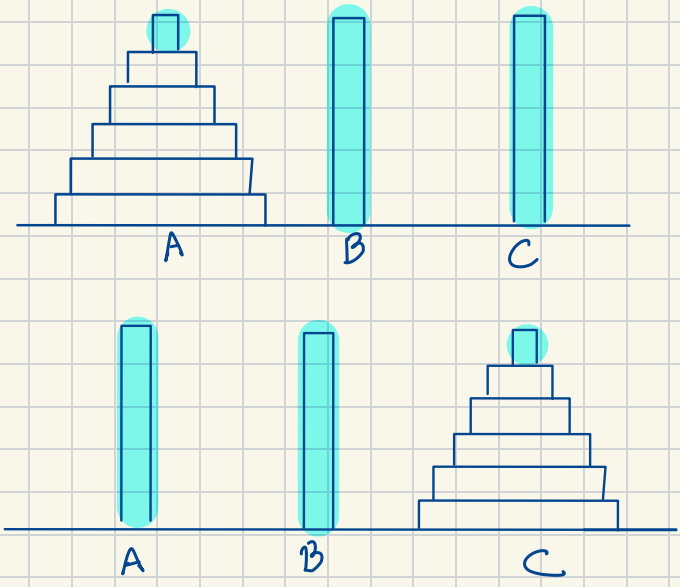
Notebook

```
9  even_member([]).
10 even_member([H|T]) :- even(H), even_member(T).
11
12 double_list([],[]).
13 double_list([H|T],[H1|T1]) :- H1 is H*2, double_list(T,T1).
14
15 even_of([],[]).
16 even_of([H|T],[H|T1]) :- even(H), even_of(T,T1).
17 even_of([_|T],T1) :- even_of(T,T1).
18
19 reverse_a_list([],[]).
20 reverse_a_list([H|T],Ans) :- reverse_a_list(T,T1), append(T1,[H],Ans).
```

เอา T ที่เหลือไป reverse

```
?- print_a_list([1,2,3,4]).
1
2
3
4
true
```

Tower of Hanoi



เสาต้น เสาปลายทาง

toh(1, A, B, _) :- write(A), write('->'), write(B), nl.

toh(N, A, B, C) :- N1 is N-1,

ย้าย 1 แผ่น (เสาพัก) toh(N1, A, C, B), toh(1, A, B), toh(N1, C, B, A)

```
1  int(1). int(2). int(3). int(4). int(5). int(6).
2
3  even(X) :- Rem is X mod 2, Rem=0.
4
5  all_even(X) :- int(X), even(X), write(X), fail.
```

```
?- all_even(X).
246false
```

ไม่ x ให้มันทำไป write

ห้าม Redo ต่อท้าย cut (!)

ปัญหาลิงหยิบกล้วย

ลิงอยู่ล่างพื้น — ถือ กล้วย

```
1  move(state(middle,onbox,middle,hasnot),grasp,state(middle,onbox,middle,has)).
2  move(state(P,onfloor,P,H),climb,state(P,onbox,P,H)).   ลิงอยู่บนกล่อง
3  move(state(P1,onfloor,P1,H),push(P1,P2),state(P2,onfloor,P2,H)).
4  move(state(P1,onfloor,B,H),walk(P1,P2),state(P2,onfloor,B,H)).
5
6
7
```

```
?- all_even(X).
2
```

```
move(state(middle,onbox,middle,hasnot),grasp,state(middle,onbox,middle,has)).
move(state(P,onfloor,P,H),climb,state(P,onbox,P,H)).
move(state(P1,onfloor,P1,H),push(P1,P2),state(P2,onfloor,P2,H)).
move(state(P1,onfloor,B,H),walk(P1,P2),state(P2,onfloor,B,H)).

depth_first(State,State,[]).
depth_first(Start,GoalState,[Operator|Ans1]) :-
      findsuccessor(StartState,Successor,Operator),
      depth_first(Successor,GoalState,Ans1),
      Ans=[Operator|Ans1].

findsuccessor(OldState,NewState,Operator) :-
      move(OldState,Operator,NewState).
```

Predicate defined in line 12

```
?- depth_first(state(atdoor,onfloor,atwindow,hasnot),state(middle,onbox,middle,has),Ans).
```

Ans = [*walk*(atdoor,atwindow), *push*(atwindow,middle), climb, grasp]

Next | 10 | 100 | 1,000 | Stop

```
print_a_list([]).

print_a_list([H|T]) :- write(H),nl,
print_a_list(T).


my_member(H,[H|_]).

my_member(H,[_|T]) :- my_member(H,T).


even(X) :- X1 is X mod 2, X1 = 0.


even_member([]).

even_member([H|T]) :- even(H),
even_member(T).


double_list([],[]).

double_list([H|T],[H1|T1]) :- H1 is H*2,
double_list(T,T1).


even_of([],[]).

even_of([H|T],[H|T1]) :- even(H),
even_of(T,T1).

even_of([_|T],T1) :- even_of(T,T1).


reverse_a_list([],[]).

reverse_a_list([H|T],Ans) :-
reverse_a_list(T,T1), append(T1,[H],Ans).


toh(1,A,B,_) :- write(A),write('-
>'),write(B),nl.

toh(N,A,B,C) :- N1 is N-
1,toh(N1,A,C,B),toh(1,A,B,C),toh(N1,C,B,A).
```