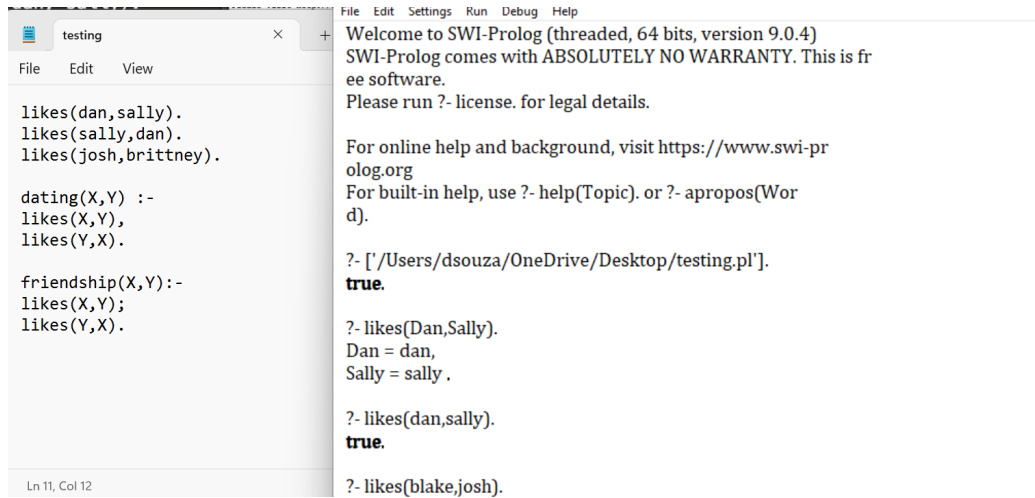


## A.I Practical - Prolog Programming



The screenshot shows the SWI-Prolog IDE with a file named 'testing'. The editor contains the following Prolog code:

```
likes(dan,sally).
likes(sally,dan).
likes(josh,brittney).

dating(X,Y) :-
likes(X,Y),
likes(Y,X).

friendship(X,Y):-
likes(X,Y);
likes(Y,X).
```

The output window shows the following interactions:

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is fr
ee software.
Please run ?- license. for legal details.

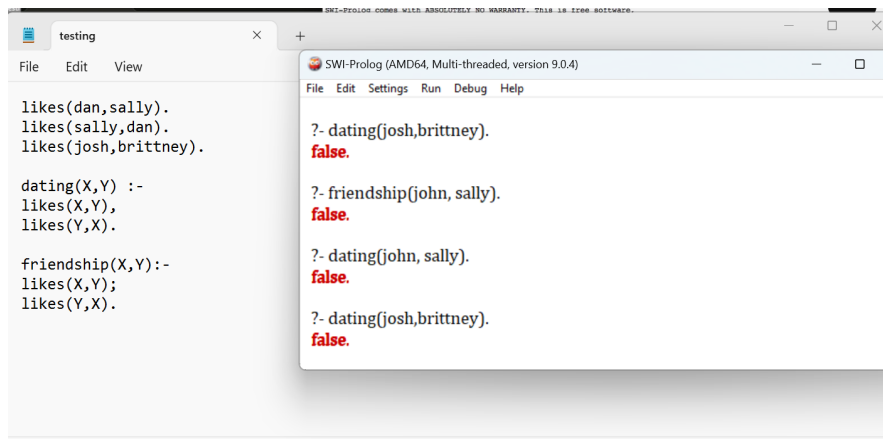
For online help and background, visit https://www.swi-pr
olog.org
For built-in help, use ?- help(Topic). or ?- apropos(Wor
d).

?- [' /Users/dsouza/OneDrive/Desktop/testing.pl'].
true.

?- likes(Dan,Sally).
Dan = dan,
Sally = sally .

?- likes(dan,sally).
true.

?- likes(blake,josh).
```



The screenshot shows the SWI-Prolog IDE with a file named 'testing'. The editor contains the following Prolog code:

```
likes(dan,sally).
likes(sally,dan).
likes(josh,brittney).

dating(X,Y) :-
likes(X,Y),
likes(Y,X).

friendship(X,Y):-
likes(X,Y);
likes(Y,X).
```

The output window shows the following interactions:

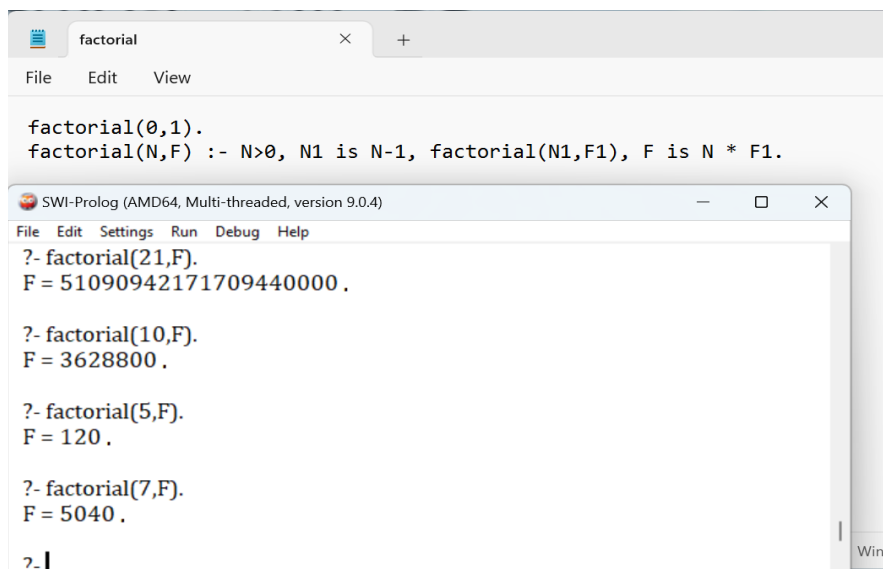
```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)

?- dating(josh,brittney).
false.

?- friendship(john, sally).
false.

?- dating(john, sally).
false.

?- dating(josh,brittney).
false.
```



The screenshot shows the SWI-Prolog IDE with a file named 'factorial'. The editor contains the following Prolog code:

```
factorial(0,1).
factorial(N,F) :- N>0, N1 is N-1, factorial(N1,F1), F is N * F1.
```

The output window shows the following interactions:

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)

?- factorial(21,F).
F = 51090942171709440000 .

?- factorial(10,F).
F = 3628800 .

?- factorial(5,F).
F = 120 .

?- factorial(7,F).
F = 5040 .

?- |
```

The screenshot shows the SWI-Prolog IDE with a file named 'sum.pl' and a console window. The code in 'sum.pl' defines a base case and a recursive case for summing a list. The console shows the execution of several queries, all returning 'true' with the correct sum.

```

sum([],0).
sum([H|T],S):-sum(T,S1),S is S1+H.

?- ['Users/dsouza/OneDrive/Desktop/sum.pl'].
true.

?- sum([1,2],S).
S = 3.

?- sum([2,4,6,8,10],S).
S = 30.

?- sum([78,1,2],S).
S = 81.

?-

```

## Program to reverse a list

The screenshot shows the SWI-Prolog IDE with a file named 'rev.pl' and a console window. The code in 'rev.pl' defines a base case and a recursive case for reversing a list. The console shows the execution of several queries, all returning 'true' with the reversed list.

```

list_concat([],L,L).
list_concat([X1|L1],L2,[X1|L3]) :-list_concat(L1,L2,L3).

list_rev([],[]).
list_rev([Head|Tail],Reversed):-
    list_rev(Tail, RevTail),list_concat(RevTail,[Head],Reversed).

Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['Users/dsouza/OneDrive/Desktop/sum.pl'].
true.

2 ?- ['Users/dsouza/OneDrive/Desktop/rev.pl'].
ERROR: c:/users/dsouza/onedrive/desktop/rev.pl:2:39: Syntax error: Operator
ERROR: c:/users/dsouza/onedrive/desktop/rev.pl:6:5: Syntax error: Operator
true.

3 ?- ['Users/dsouza/OneDrive/Desktop/rev.pl'].
true.

3 ?- list_rev([a,b,c,d,e],NewList).
NewList = [e, d, c, b, a].

4 ?-

```

## Program to find the length of a list

The screenshot shows the SWI-Prolog IDE with a file named 'lengthlist.pl' and a console window. The code in 'lengthlist.pl' defines a base case and a recursive case for finding the length of a list. The console shows the execution of several queries, all returning 'true' with the correct length.

```

list_length([],0).
list_length([_|TAIL],N):-list_length(TAIL,N1),N is N1+1.

ERROR:      list_length/2
false.

6 ?- list_length([a,b,c,d,e],Len).
Len = 5.

7 ?- list_length([2,3,4,5,6,7,8,7],Len).
Len = 8.

8 ?-

```

## Find max between to numbers

The screenshot shows a Prolog IDE with two windows. The left window, titled 'findmax', contains the following code:

```
find_max(X,Y,X):-X>=Y,!.  
find_max(X,Y,Y):-X<Y.
```

The right window, titled 'SWI-Prolog (console)', shows the following output:

```
false.  
6 ?- list_length([a,b,c,d,e],Len).  
Len = 5.  
7 ?- list_length([2,3,4,5,6,7,8,7],Len).  
Len = 8.  
8 ?- ['./Users/dsouza/OneDrive/Desktop/findmax.pl'].  
true.  
9 ?- find_max(2934,2424,Max).  
Max = 2934.  
10 ?- |
```

## Find nth fibonacci of a number

The screenshot shows a Prolog IDE with two windows. The left window, titled 'findnthfib', contains the following code:

```
fib(1,0).  
fib(2,1).  
fib(X,Y):-X1 is X-1, fib(X1,Y1),X2 is X-2, fib(X2,Y2),Y is Y1+Y2.
```

The right window, titled 'SWI-Prolog (console)', shows the following output:

```
9 ?- find_max(2934,2424,Max).  
Max = 2934.  
10 ?- ['./Users/dsouza/OneDrive/Desktop/findnthfib.pl'].  
true.  
11 ?- fib(5,X).  
X = 3 .  
12 ?- fib(6,Y).  
Y = 5 |
```