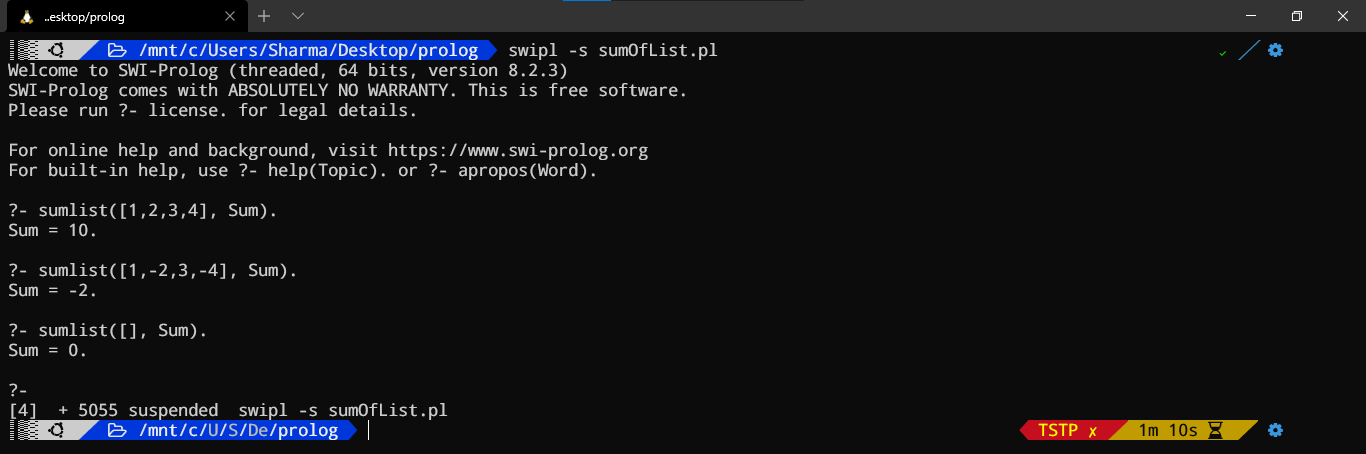
1. **Write a prolog  program to compute the sum of the list.**

CODE:

sumlist([], 0).

sumlist([H|T], Sum) :- sumlist(T, N1), Sum is N1+H.

OUTPUT:



1. **Write a prolog program to find fibonacci series.**

CODE:

fib\_seq(0,[0]).

fib\_seq(1,[0,1]).

fib\_seq(N,Seq) :-

N > 1,

fib\_seq\_(N,SeqR,1,[1,0]),

reverse(SeqR,Seq).

fib\_seq\_(N,Seq,N,Seq).

fib\_seq\_(N,Seq,N0,[B,A|Fs]) :-

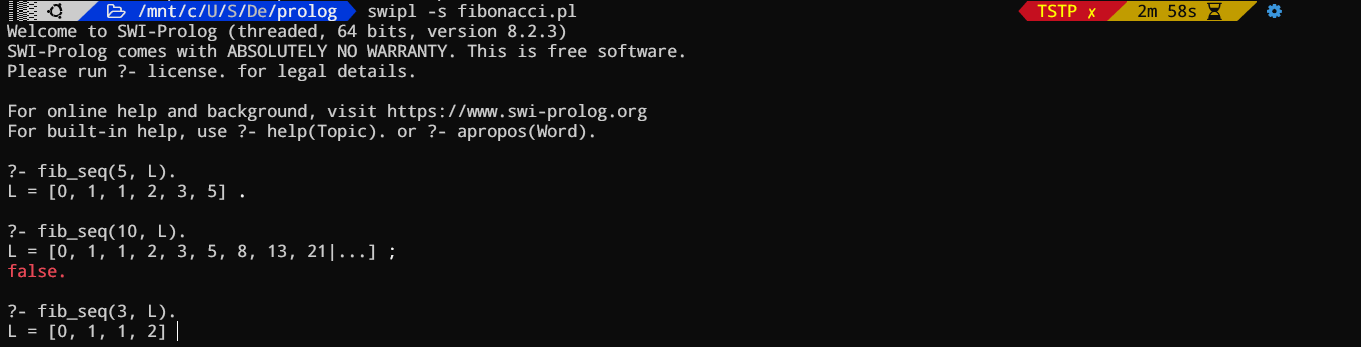
N > N0,

N1 is N0+1,

C is A+B,

fib\_seq\_(N,Seq,N1,[C,B,A|Fs]).

OUTPUT:



1. **Write a prolog program to union of two list X and Y.**

CODE:

union([],[],[]).

union(List1,[],List1).

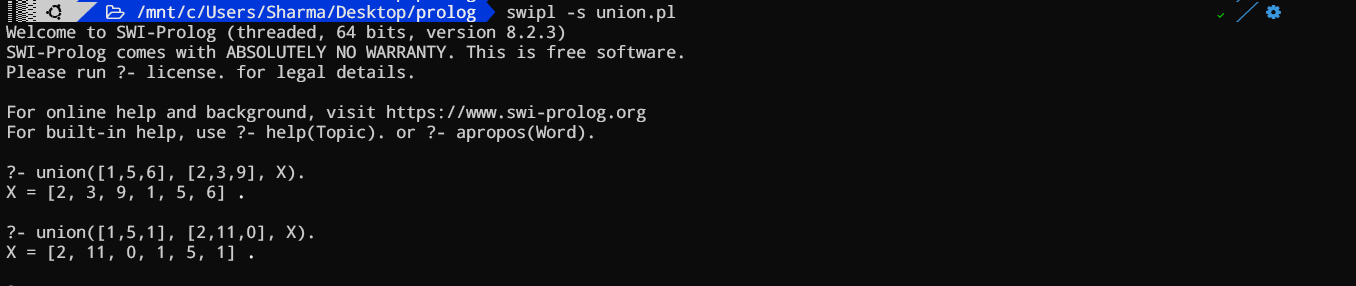
union(List1, [Head2|Tail2], [Head2|Output]):-

\+(member(Head2,List1)), union(List1,Tail2,Output).

union(List1, [Head2|Tail2], Output):-

member(Head2,List1), union(List1,Tail2,Output).

OUTPUT:



1. **Write a prolog program to divide a list in two list which are appropriately of same length.**

CODE:

div([],[],[]).

div([X],[X],[]).

div([X,Y|List],[X|List1],[Y|List2]):- div(List,List1,List2).

OUTPUT:

