**Ques 2. Write a Prolog program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.**

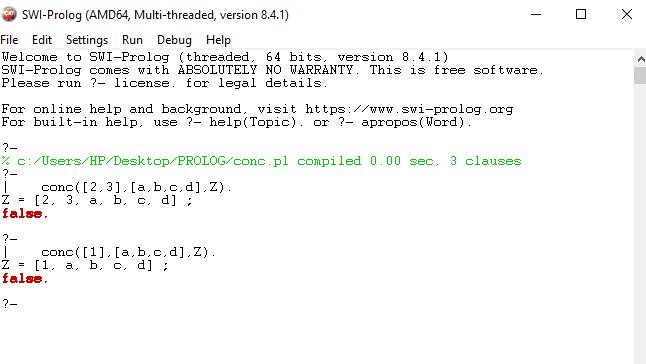
**Code**

conc([],L2,L2).

conc(L1,[],L1).

conc([X|L1],L2,[X|L3]):-conc(L1,L2,3).

**Output**



**Ques 3. Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.**

**Code**

conc([],L2,L2).

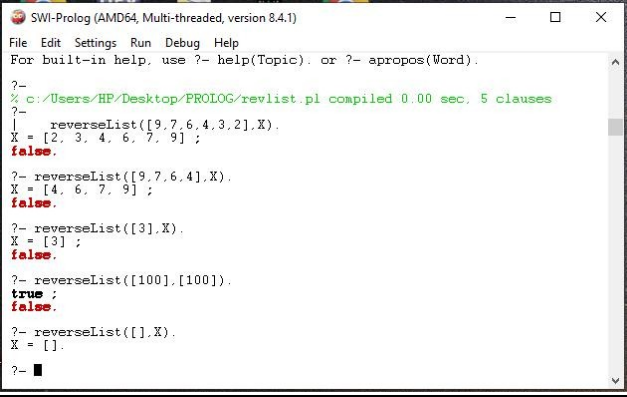
conc(L1,[],L1).

conc([X|L1],L2,[X|L3]):-conc(L1,L2,L3).

reverseList([],[]).

reverseList([H|T],R):- reverseList(T,X),conc(X,[H],R).

**Output**



**Ques 14. Write a prolog program to implement insert\_nth (I, N, L, R) that inserts an item I into Nth position of list L to generate a list R.**

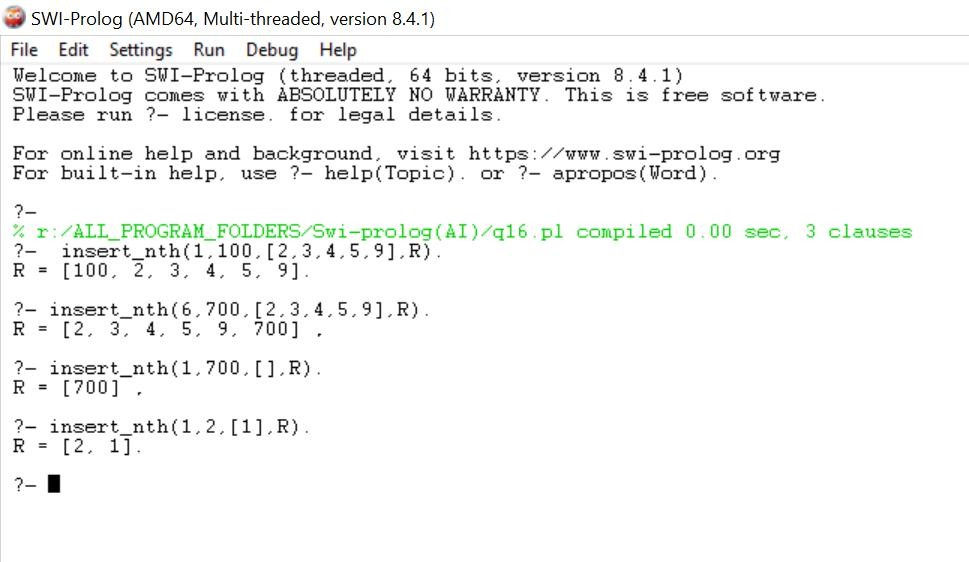
**Code**

insert\_nth(1,N,[],[N]).

insert\_nth(I,N,[H|T],[H|Y]):- I>1, I1 is I-1, insert\_nth(I1,N,T,Y).

insert\_nth(1,N,[H|T],[N,H|T]).

**Output**



**Ques 15. Write a Prolog program to implement delete\_nth (N, L, R) that removes the element on Nth position from a list L to generate a list R.**

**Code**

delete\_nth(N,[],[]). delete\_nth(N,[H|T],[H|Y]):- N>1, N1 is N-1, delete\_nth(N1,T,Y).

delete\_nth(1,[H|T],T).

**Output**

