U18CO018

Shubham Shekhaliya

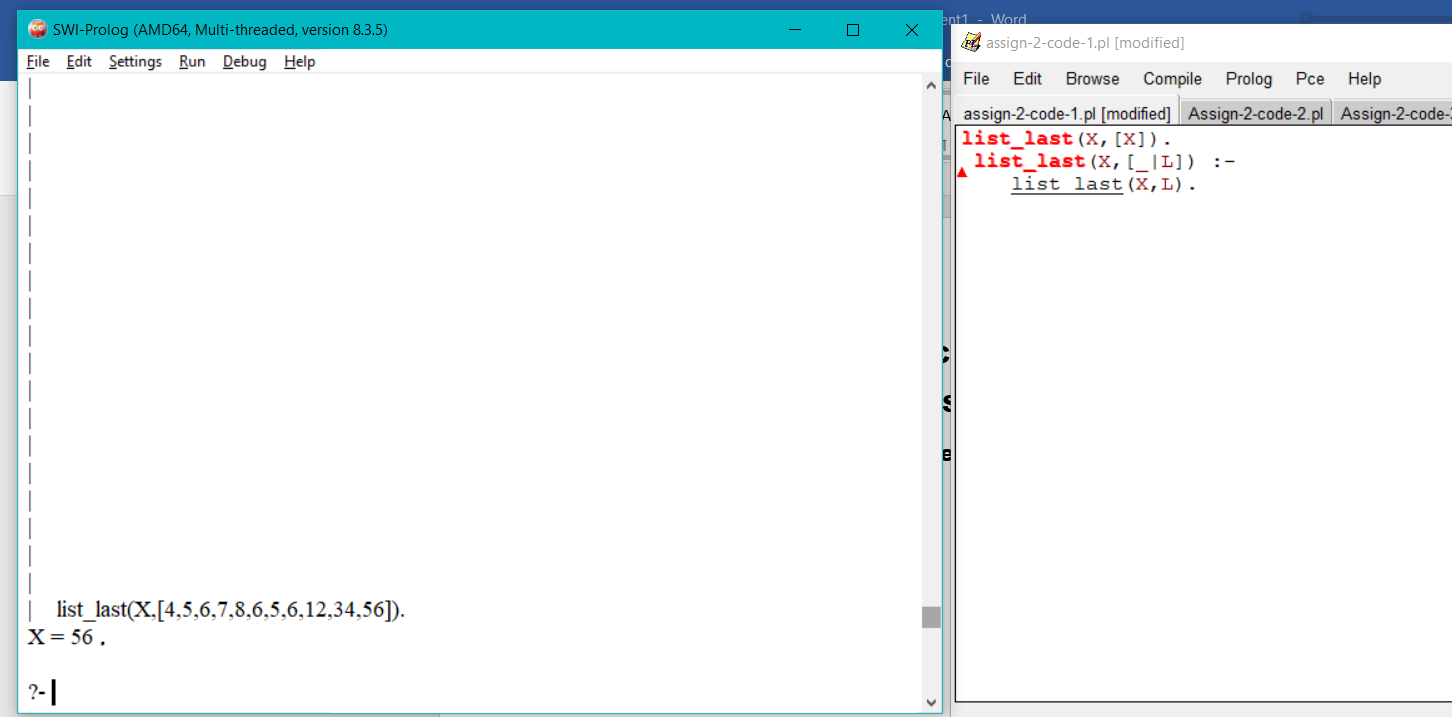
Assignment-2 (AIML)

**1-> Find the last element of a list.**

**list\_last(X,[X]).**

**list\_last(X,[\_|L]) :-**

**list\_last(X,L).**

****

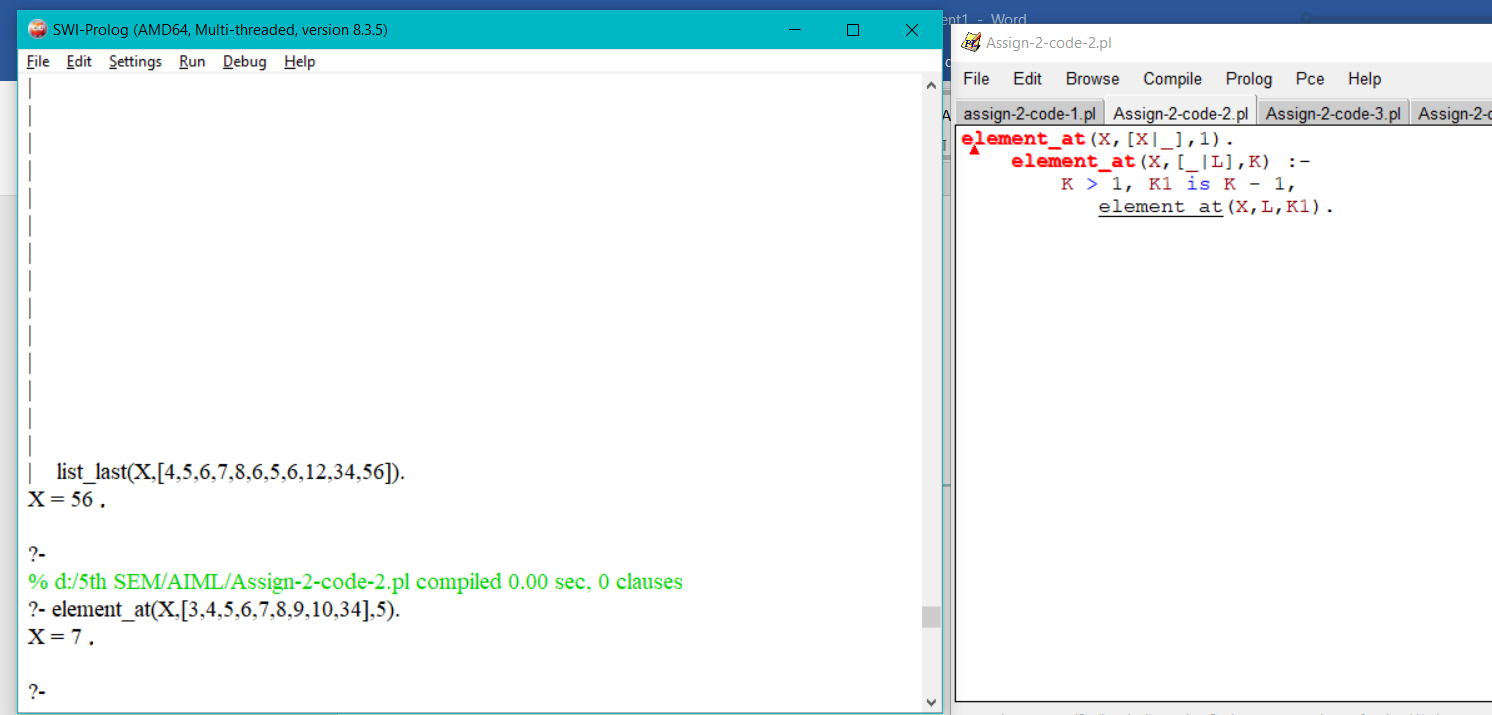
**2-> Find the K'th element of a list.**

**element\_at(X,[X|\_],1).**

**element\_at(X,[\_|L],K) :-**

**K > 1, K1 is K - 1,**

**element\_at(X,L,K1).**

****

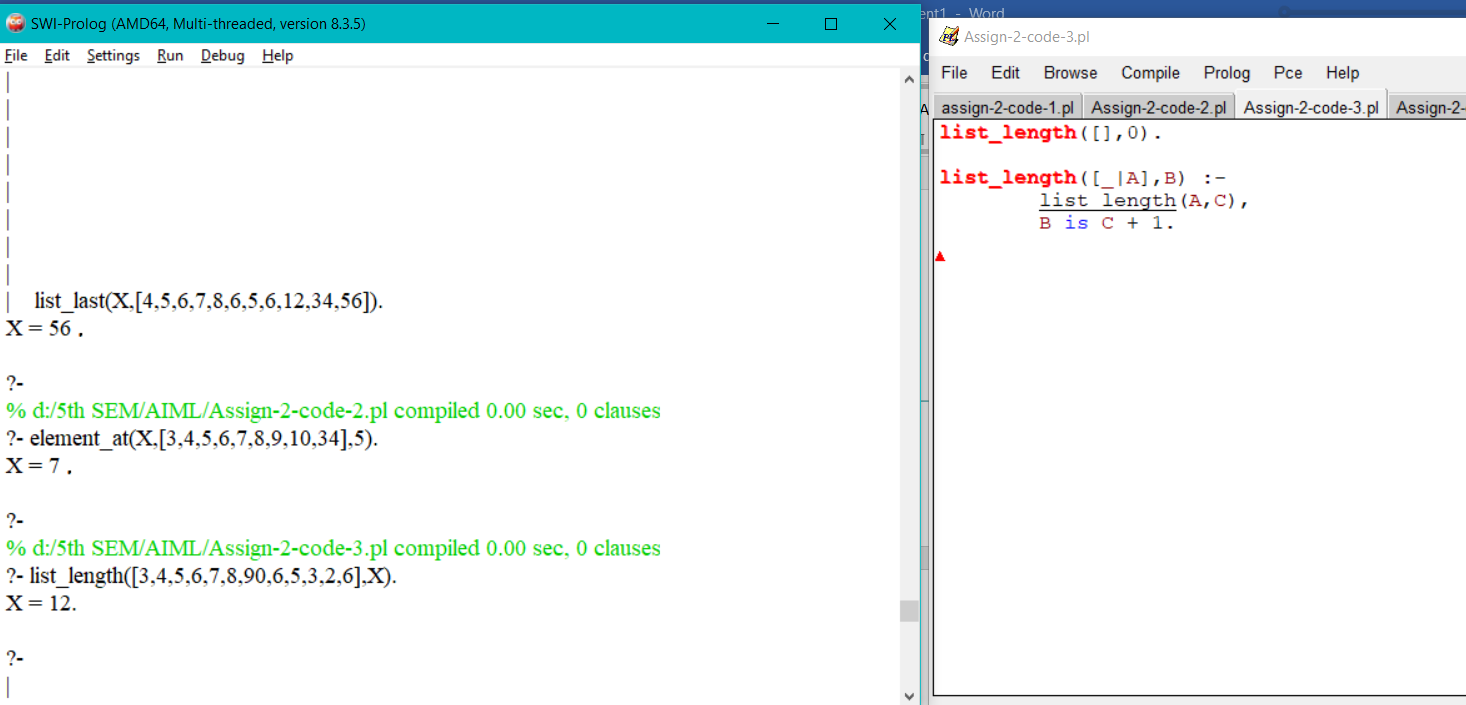
**3-> Find the number of elements in list.**

**list\_length([],0).**

**list\_length([\_|A],B) :-**

**list\_length(A,C),**

**B is C + 1.**

****

**4-> Find out whether a list is a palindrome.**

**concate([],List,List).**

**concate([X|List1],List2,[X|List3]) :-**

**concate(List1,List2,List3).**

**rev([],[]).**

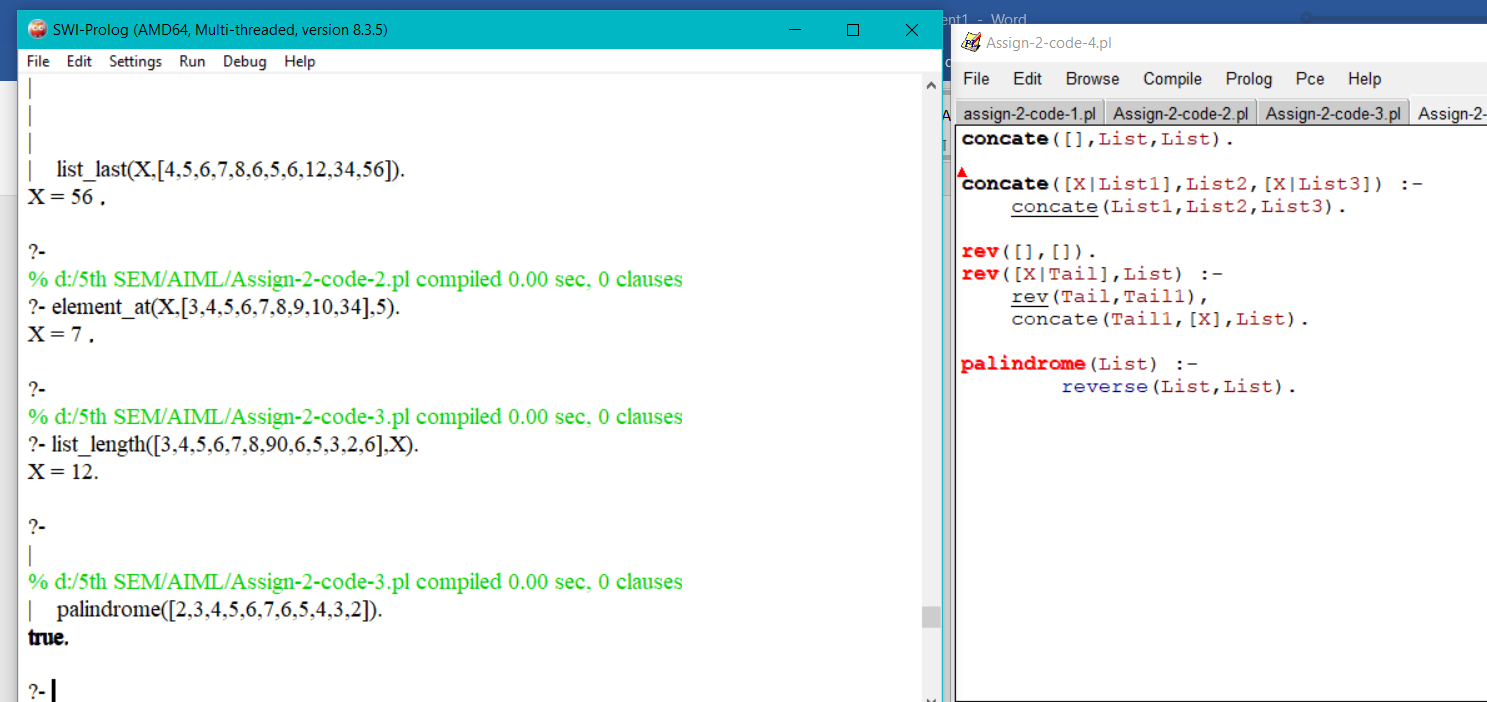
**rev([X|Tail],List) :-**

**rev(Tail,Tail1),**

**concate(Tail1,[X],List).**

**palindrome(List) :-**

**reverse(List,List).**

****

**5-> Eliminate consecutive duplicates of list elements.**

**compress([],[]).**

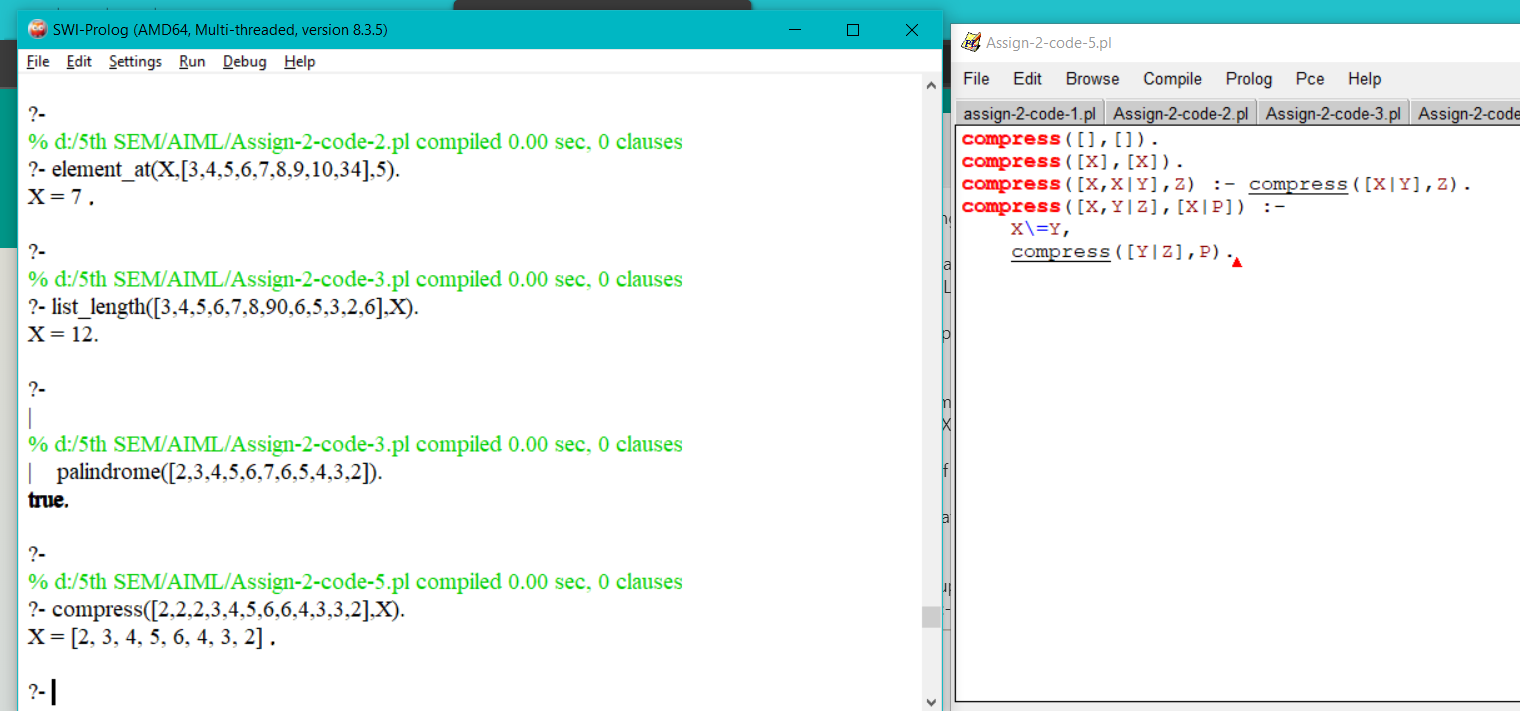
**compress([X],[X]).**

**compress([X,X|Y],Z) :- compress([X|Y],Z).**

**compress([X,Y|Z],[X|P]) :-**

**X\=Y,**

**compress([Y|Z],P).**

****

**6-> Duplicate the elements of a list a given number of times.**

**duplicate([H], 1, [H]).**

**duplicate([H], N, [H|X]) :-**

**M is N - 1,**

**M > 0,**

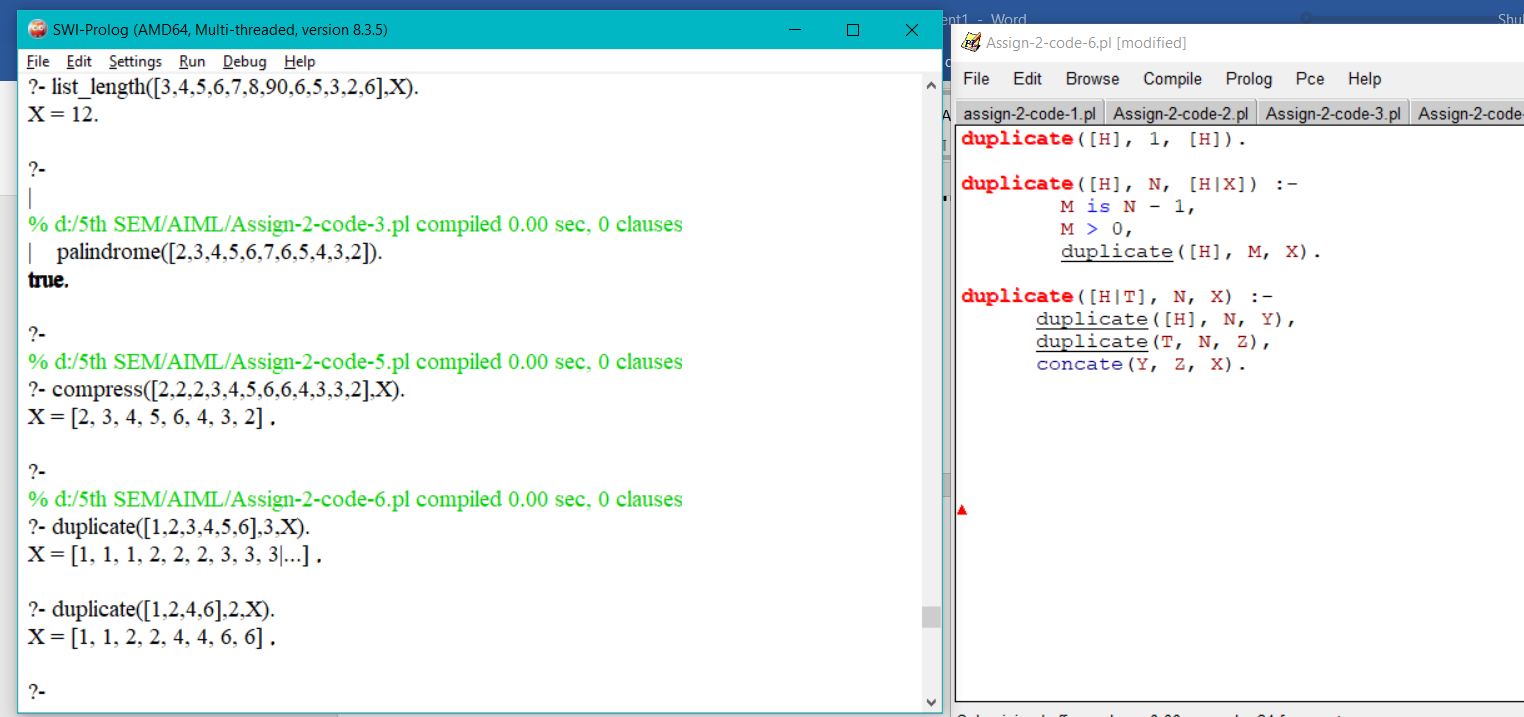
**duplicate([H], M, X).**

**duplicate([H|T], N, X) :-**

**duplicate([H], N, Y),**

**duplicate(T, N, Z),**

**concate(Y, Z, X).**

****

**7-> Drop every N'th element from a list.**

**drop(L1,N,L2) :- drop(L1,N,L2,N).**

**drop([],,[],).**

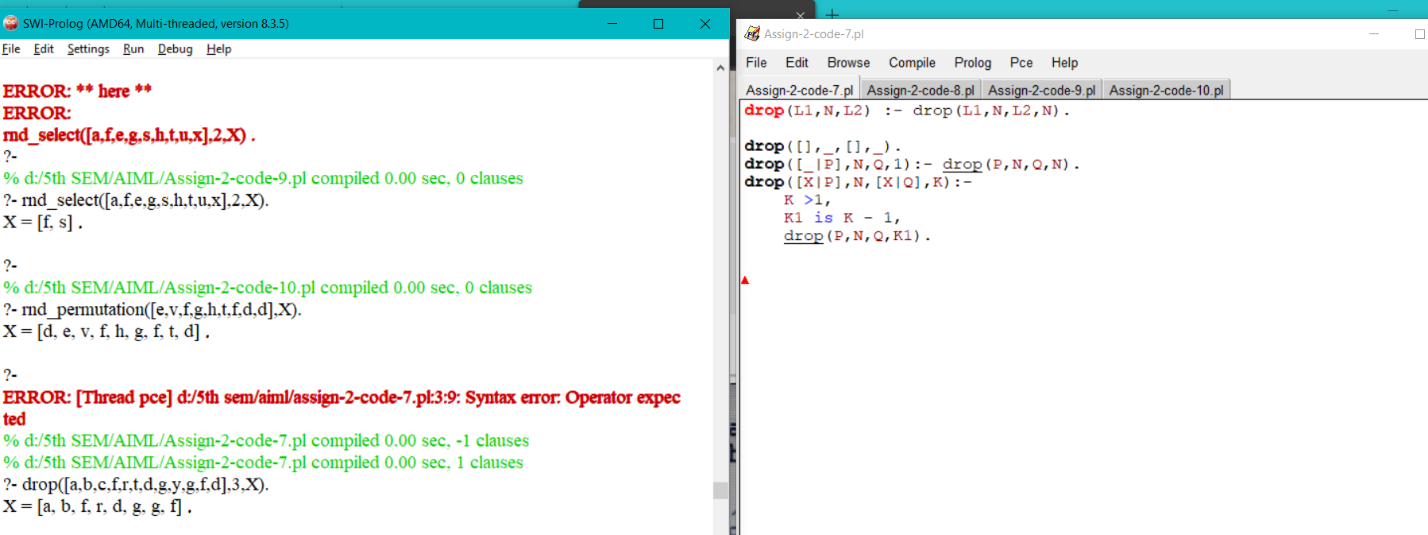
**drop([\_|P],N,Q,1):- drop(P,N,Q,N).**

**drop([X|P],N,[X|Q],K):-**

**K >1,**

**K1 is K - 1,**

**drop(P,N,Q,K1).**

****

**8-> Rotate a list N places to the left.**

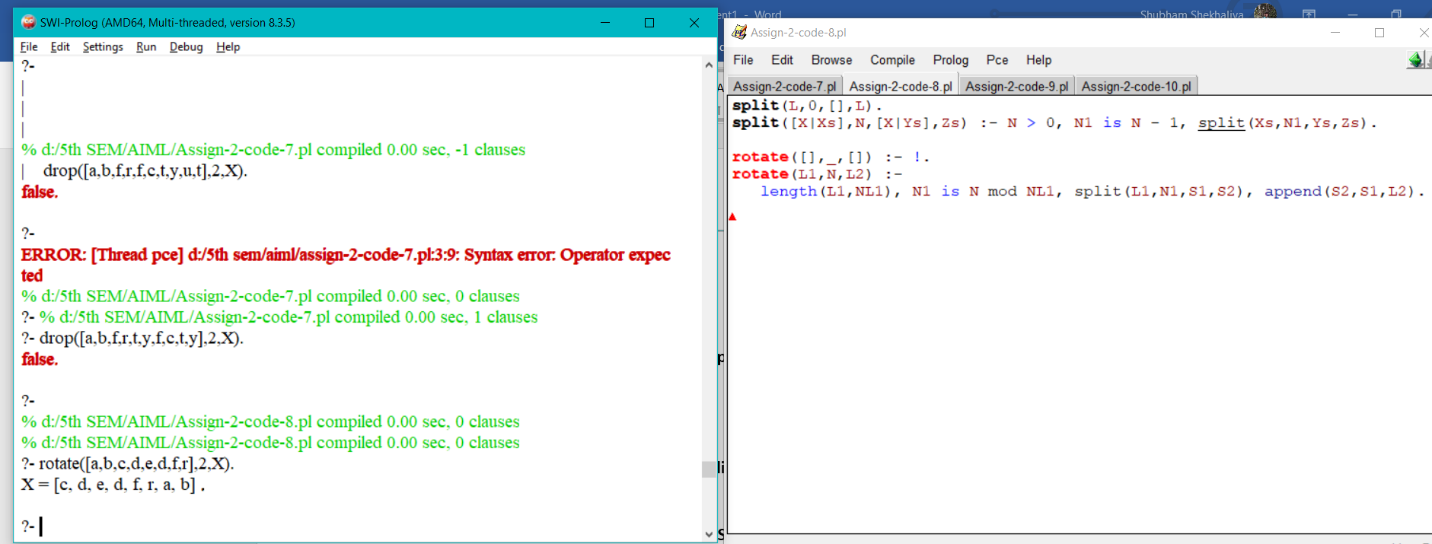
**split(L,0,[],L).**

**split([X|Xs],N,[X|Ys],Zs) :- N > 0, N1 is N - 1, split(Xs,N1,Ys,Zs).**

**rotate([],\_,[]) :- !.**

**rotate(L1,N,L2) :-**

**length(L1,NL1), N1 is N mod NL1, split(L1,N1,S1,S2), append(S2,S1,L2).**

****

**9-> Extract a given number of randomly selected elements from a list.**

**remove\_at(X,[X|Xs],1,Xs).**

**remove\_at(X,[Y|Xs],K,[Y|Ys]) :- K > 1,**

**K1 is K - 1, remove\_at(X,Xs,K1,Ys).**

**rnd\_select(\_,0,[]).**

**rnd\_select(Xs,N,[X|Zs]) :- N > 0,**

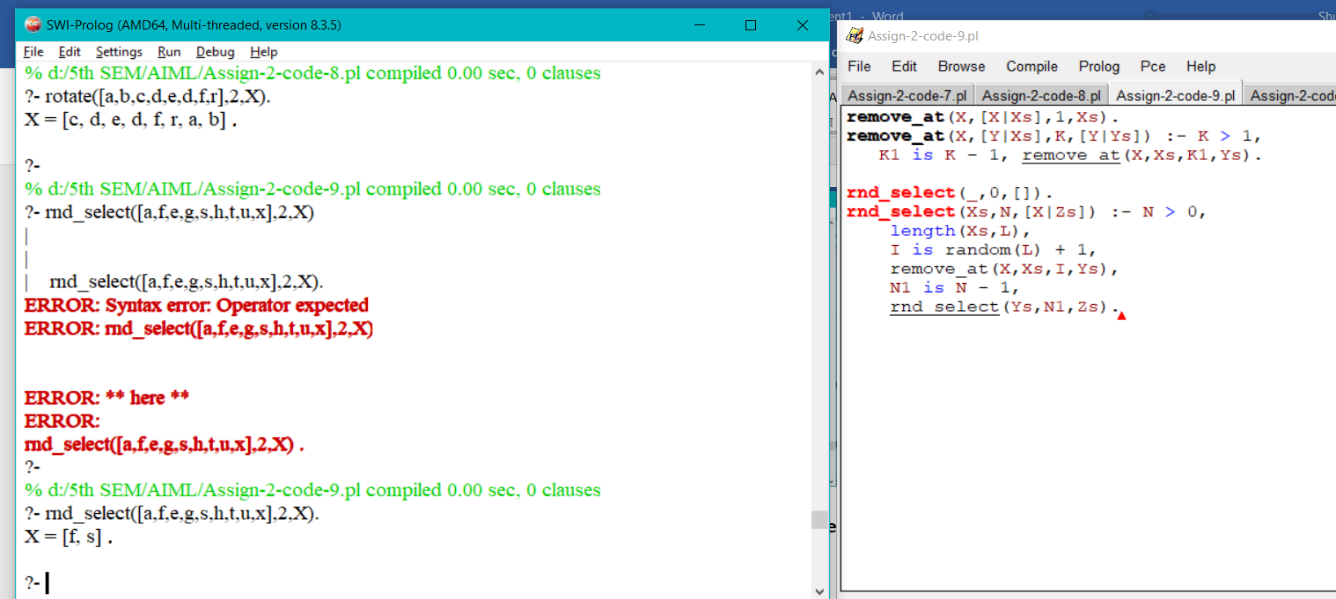
**length(Xs,L),**

**I is random(L) + 1,**

**remove\_at(X,Xs,I,Ys),**

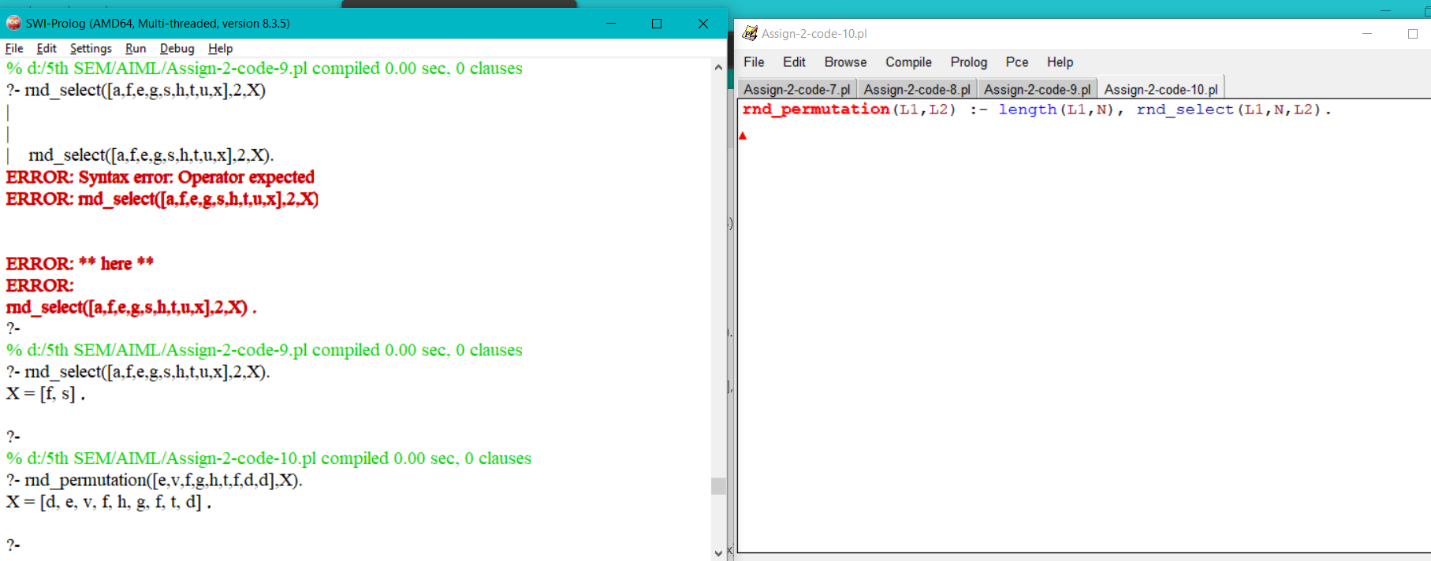
**N1 is N - 1,**

**rnd\_select(Ys,N1,Zs).**

****

**10-> Generate a random permutation of the elements of a list.**

**rnd\_permutation(L1,L2) :- length(L1,N), rnd\_select(L1,N,L2).**

****