Name: ROHAN BHALLA
Course: B. Sc. (H) Computer Science, III Year, VI Semester
College Roll No. : CSC/20/51
University Roll No. : 20059570041
SUBJECT: PRACTICAL FILE for Core Paper XIII: Artificial Intelligence

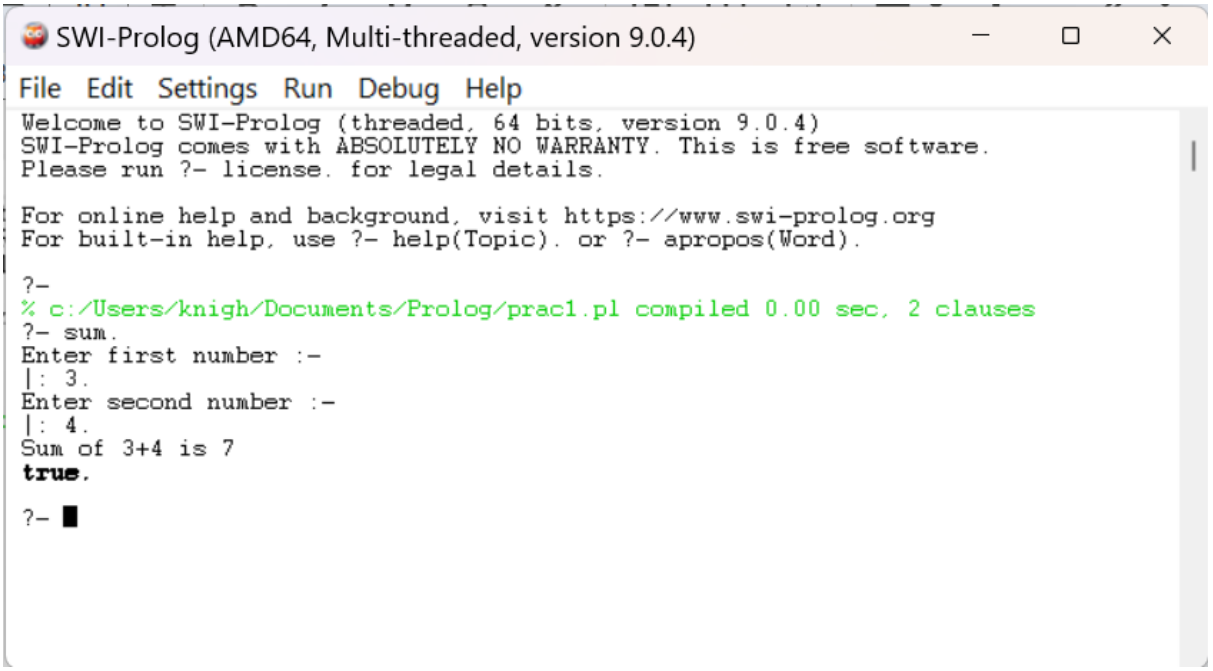
INDEX

Q1. Write a Prolog Program to calculate the sum of two numbers.	4
Q2. Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.	5
Q3. Write a Prolog Program to implement factorial (N, F) where F represents the factorial of a number N.	6
Q4. Write a Prolog Program to implement generate_fib(N,T) where T represents the Nth term of the Fibonacci Series.	7
Q5. Write a Prolog program to implement GCD of two numbers.	8
Q6. Write a Prolog Program to implement power (Num,Pow,) : where Num is raised to the power Pow to get.	9
Q7. Write a Prolog Program to implement multi (N1, N2, R) : where N1 and N2 denote the numbers to be multiplied and R represents the result.	10
Q8. Write a Prolog Program to implement memb(X, L): to check whether X is a member of L or not.	11
Q9. Write a Prolog Program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulting list L3.	12
Q10. Write a Prolog Program to implement reverse (L, R) where List L is original and List R is reversed list.	13
Q11. Write a Prolog Program to implement palindrome (L) which checks whether a list L is a palindrome or not.	14
Q12. Write a Prolog Program to implement sumlist(L, S) so that S is the sum of a given list L	15
Q13. Write a Prolog Program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.	16
Q14. Write a Prolog Program to implement nth_element (N, L, X) where N is the desired position, L is a list and X represents the Nth element of L.	17
Q15. Write a Prolog Program to implement maxlist(L, M) so that M is the maximum number in the list.	18
Q16. 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.	19

Q17. 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.	20
Q18. Write a Prolog Program to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.	21

Q1. Write a Prolog Program to calculate the sum of two numbers.
Ans:

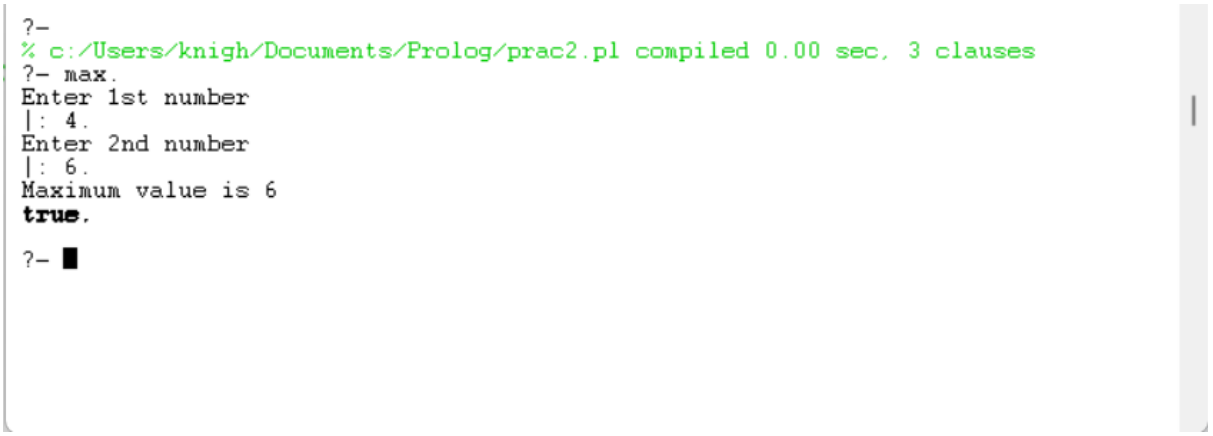
```
sum(A,B):- Z is A+B,write(Z).
sum:-
   write('Enter first number :- '),nl,read(A),
   write('Enter second number :- '),nl,read(B),
   write('Sum of '),write(A+B),write(' is '),sum(A,B).
```



Q2. Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.

Ans:

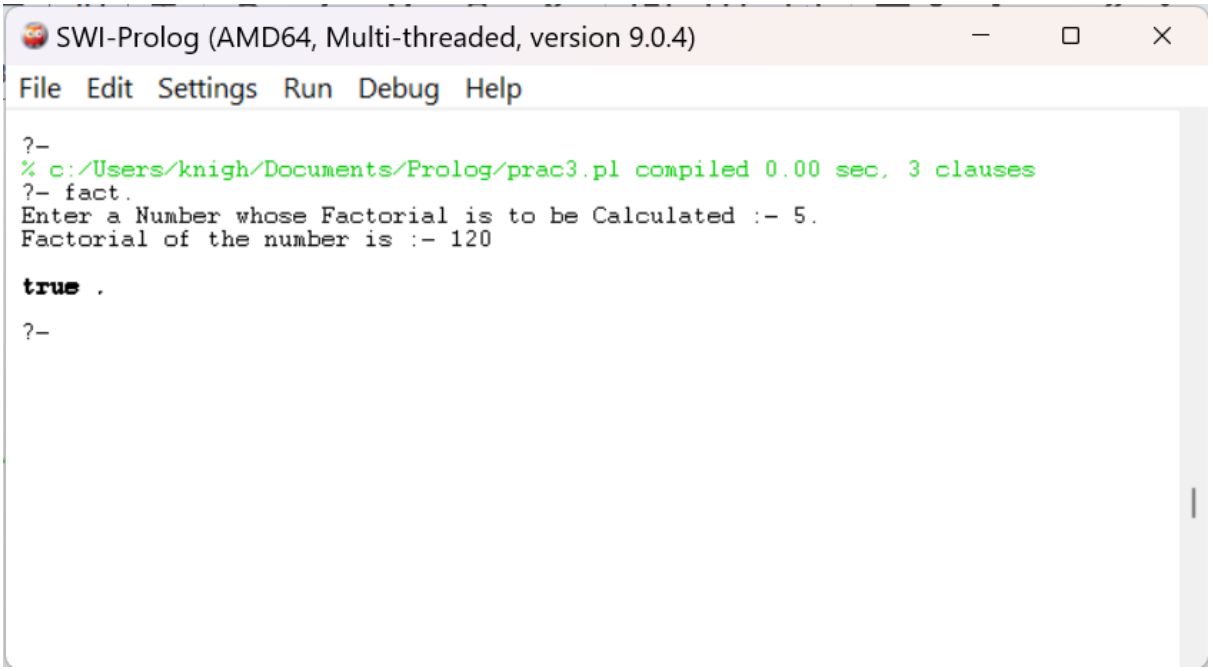
```
maxmax(A,B):- A>=B,write(A).
max(A,B):- B>A,write(B).
max:-
   write('Enter 1st number '),nl,read(A),
   write('Enter 2nd number '),nl,read(B),
   write('Maximum value is '),max(A,B).
```



Q3. Write a Prolog Program to implement factorial (N, F) where F represents the factorial of a number N.

Ans:

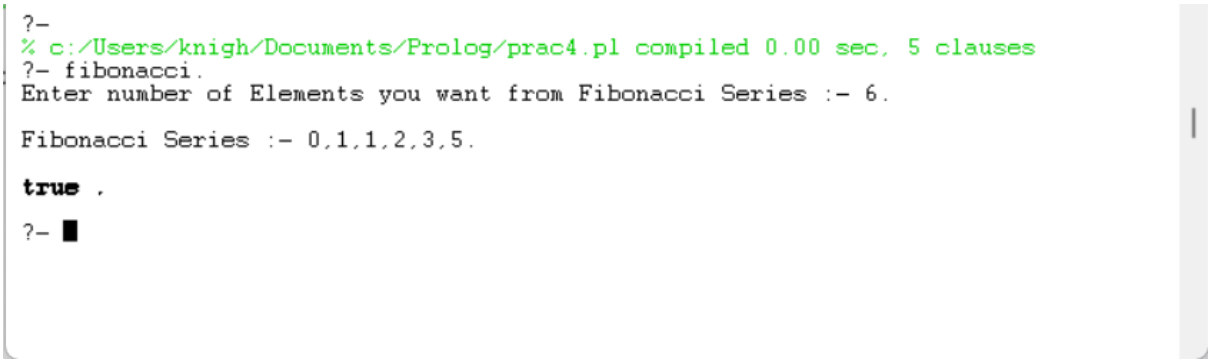
```
show(1,Z):- write('Factorial of the number is :- '),write(Z),nl.
show(X,Z):-
    Temp is Z*X,
    Y is X-1,
    show(Y,Temp).
fact :- write('Enter a Number whose Factorial is to be Calculated :- '),read(A),
show(A,1),nl.
```



Q4. Write a Prolog Program to implement generate_fib(N,T) where T represents the Nth term of the Fibonacci Series.

Ans:

```
print_fibo(_A,_B,_C,L):- L=:=1,write('0.').
print_fibo(_A,_B,_C,L):- L=:=0,write(' - '),nl.
print_fibo(A,_B,C,_L):- C=:=1,write(A),write('.'),nl.
print_fibo(A,B,C,L):- write(A),write(','),
   E is A+B,
   D is B,
   F is C-1,print_fibo(D,E,F,L).
fibonacci:-
   write('Enter number of Elements you want from Fibonacci Series :- '),read(A),nl,
   write('Fibonacci Series :- '),print_fibo(0,1,A,A),nl.
```



Q5. Write a Prolog program to implement GCD of two numbers.

Ans:

```
gcd(0,_X):-write("GCD is 0").
gcd(_X,0):-write("GCD is 0").
gcd(1,_X):-write("GCD is 1").
gcd(_X,1):-write("GCD is 1").
gcd(A,B):-startnow(A,B,2).
confirm(0,0,_A,_B,C):-write("GCD is :- "),write(C).
confirm(_X,_Y,A,B,C):- C<A,S is C+1,startnow(A,B,S).
startnow(A,B,C):-X is A mod C,Y is B mod C,confirm(X,Y,A,B,C).
gcd:-
   write("Enter 1st number(Smaller) :- "),read(A),nl,
   write("Enter 2nd number(Bigger) :- "),read(B),nl,
   write("GCD is  :- "),gcd(A,B),nl.
```

```
?-
% c:/Users/knigh/Documents/Prolog/prac5.pl compiled 0.00 sec, 0 clauses
?- gcd.
Enter 1st number(Smaller) :- 5.

Enter 2nd number(Bigger) :- |: 25.

GCD is  :- GCD is :- 5
true .
```

Q6. Write a Prolog Program to implement power (Num,Pow,) : where Num is raised to the power Pow to get.

Ans:

```
calc_power(A,B):- Z is A^B,write(Z),nl.

base:-
   write('Enter Base Value :- '),read(Num),nl,
   write('Enter Power Value :- '),read(Pow),nl,
   write('Resultant Value :- '),calc_power(Num,Pow),nl.
```

```
?-
% c:/Users/knigh/Documents/Prolog/prac6.pl compiled 0.00 sec, 2 clauses
?- base.
Enter Base Value :- 2.

Enter Power Value :- |: 4.

Resultant Value :- 16

true.

?-
```

Q7. Write a Prolog Program to implement multi (N1, N2, R) : where N1 and N2 denote the numbers to be multiplied and R represents the result.

Ans:

```
mul(A,B,_Z):- Temp is A*B,write(Temp),nl.
mul:-
   write('Enter 1st number :- '),read(A),nl,
   write('Enter 2nd number :- '),read(B),nl,
   write('Mulitplication Result :-  '),mul(A,B,1).
```

```
?-
% c:/Users/knigh/Documents/Prolog/prac7.pl compiled 0.00 sec, 2 clauses
?- mul.
Enter 1st number :- 4.

Enter 2nd number :- |: 6.

Mulitplication Result :-  24
true.

?-
```

Q8. Write a Prolog Program to implement memb(X, L): to check whether X is a member of L or not.
Ans:

```
list([1,2,3,4,5,6,7,8,9,10]).
check(X):-list(L),member(X,L).
findinlist:-
    write("Enter Number to check in Given List (1,2,3,4,5,6,7,8,9,10) :-  "),read(A),check(A),nl.
```

```
?-
% c:/Users/knigh/Documents/Prolog/prac8.pl compiled 0.00 sec, 3 clauses
?- findinlist.
Enter Number to check in Given List (1,2,3,4,5,6,7,8,9,10) :-  8.

true .

?-
```

Q9. Write a Prolog Program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulting list L3.

Ans:

```
conc([], L, L).
conc([H|T], L2, [H|L3]):- conc(T, L2, L3).

concat :-
    write('Enter first list: '),
    read(List1),
    write('Enter second list: '),
    read(List2),
    conc(List1, List2, Result),
    write('Concatenated list: '),
    write(Result).
```

```
?-
% c:/Users/knigh/Documents/Prolog/prac9.pl compiled 0.00 sec, 3 clauses
?- concat.
Enter first list: (1,6,3,2).
Enter second list: |: (1,5,2).

false.

?- concat.
Enter first list: [1,2,3].
Enter second list: |: [4,5,6].
Concatenated list: [1,2,3,4,5,6]
true.

?-
```

Q10. Write a Prolog Program to implement reverse (L, R) where List L is original and List R is reversed list.

Ans:

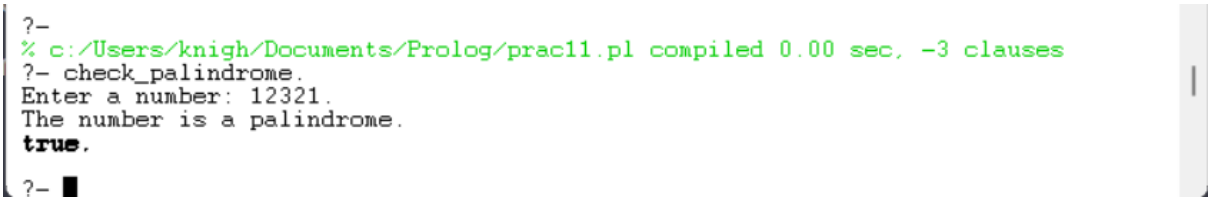
```
reverse([], []).
reverse([H|T], R) :- reverse(T, RevT), append(RevT, [H], R).

reverse_list :-
    write('Enter a list: '),
    read(List),
    reverse(List, RevList),
    write('Original List: '), write(List), nl,
    write('Reversed List: '), write(RevList), nl.
```

```
?-
% c:/Users/knigh/Documents/Prolog/prac10.pl compiled 0.00 sec, 3 clauses
?- reverse_list.
Enter a list: [1,2,3,4,5].
Original List: [1,2,3,4,5]
Reversed List: [5,4,3,2,1]
true.

?-
```

Q11. Write a Prolog Program to implement palindrome (L) which checks whether a list L is a palindrome or not.

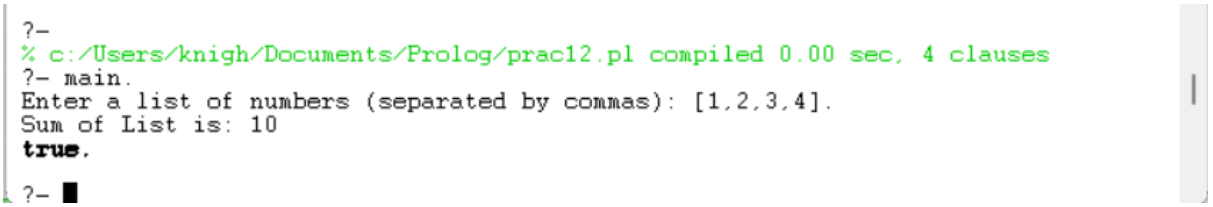
Ans:

```
palindrome(Number) :-
    number_chars(Number, Digits),
    reverse(Digits, ReversedDigits),
    Digits = ReversedDigits.

check_palindrome :-
    write('Enter a number: '),
    read(Number),
    (palindrome(Number)
    -> write('The number is a palindrome.')
    ; write('The number is not a palindrome.')
    ).
```



Q12. Write a Prolog Program to implement sumlist(L, S) so that S is the sum of a given list L
Ans:

```
sumlist(List) :- sumlist(List, 0).
sumlist([H|T], S) :-
    C is H + S,
    sumlist(T, C).
sumlist([], S) :-
    write("Sum of List is: "),
    write(S),
    nl.
main :-
    write("Enter a list of numbers (separated by commas): "),
    read(List),
    sumlist(List).
```



Q13. Write a Prolog Program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.

Ans:

```
len([], 0).
len([_H | T], A) :- len(T, A1), A is A1 + 1.
evenlength :-
    write("Enter list elements: "), read(L),
    len(L, Len), S is Len mod 2, checkeven(S).
checkeven(X) :- X =:= 0, write("List is of Even Length :- True").
checkeven(_) :- write("List is of Even Length :- False").
oddlength :-
    write("Enter list elements: "), read(L),
    len(L, Len), S is Len mod 2, checkodd(S).
checkodd(X) :- X =:= 1, write("List is of Odd Length :- True").
```

```
?-
% c:/Users/knigh/Documents/Prolog/prac13.pl compiled 0.00 sec, 7 clauses
?- evelength.
Correct to: "evenlength"?
Please answer 'y' or 'n'? yes
Enter list elements: .

ERROR: Stream user_input:168:22 Syntax error: Unexpected end of clause
?- evenlength.
Enter list elements: [1,2,3,4].
List is of Even Length :- True
true .

?- evenlength.
Enter list elements: [1,2,3].
List is of Even Length :- False
true.

?-
```

Q14. Write a Prolog Program to implement nth_element (N, L, X) where N is the desired position, L is a list and X represents the Nth element of L.

Ans:

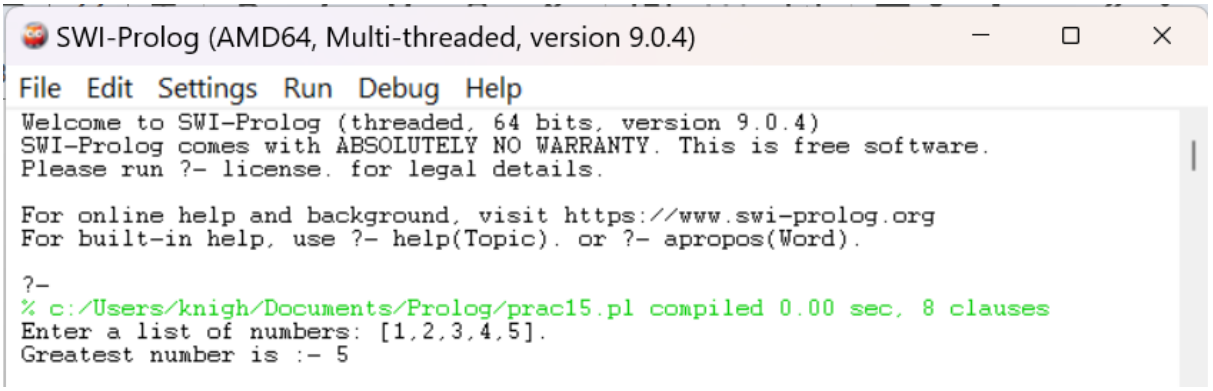
```
nth_element(1, [H|_], H).
nth_element(N, [_|T], X) :- N > 1, N1 is N-1, nth_element(N1, T, X).

main :-
    write('Enter a list: '),
    read(List),
    write('Enter the position: '),
    read(Position),
    nth_element(Position, List, X),
    write('Element at position '), write(Position), write(' is '), write(X).
```

```
% c:/Users/knigh/Documents/Prolog/prac14.pl compiled 0.00 sec, 4 clauses
?- main.
Enter a list: [1,2,3,4].
Enter the position: |: 2.
Element at position 2 is 2
true .

?-
```

Q15. Write a Prolog Program to implement maxlist(L, M) so that M is the maximum number in the list.

Ans:

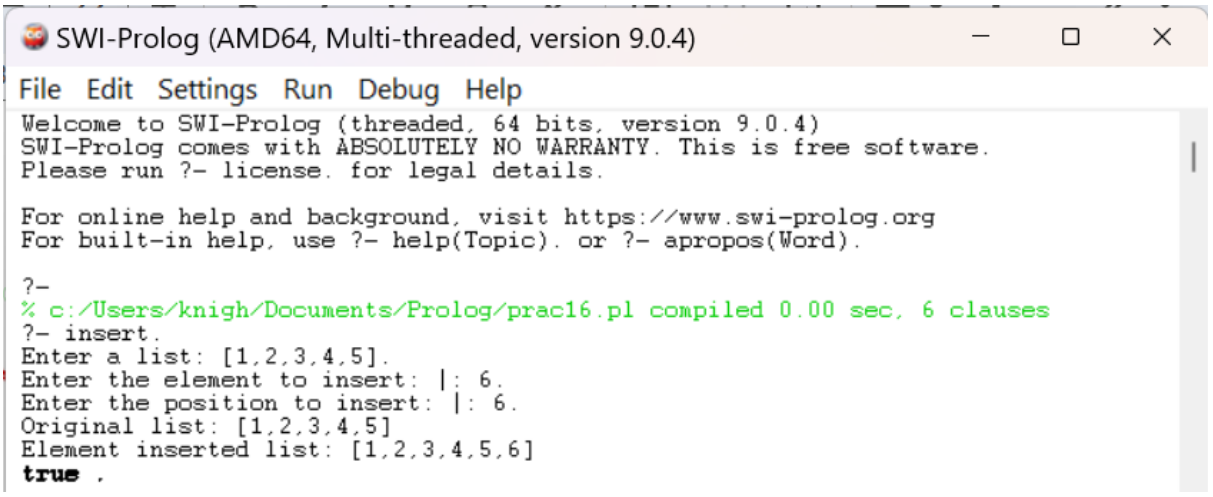
```
max([H|T],M):- H>M,max(T,H).
max([H|T],M):- H=<M,max(T,M).
max([],M):- M>=0,write("Greatest number is :- "),write(M),nl.
maxList([]):-write("List can't be Empty").
maxList([H|T]):- M is H,max(T,M),nl.
get_list(List) :-
    write('Enter a list of numbers: '),
    read(List).
:- initialization(main).
main :-
    get_list(List),
    maxList(List).
```



Q16. 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.

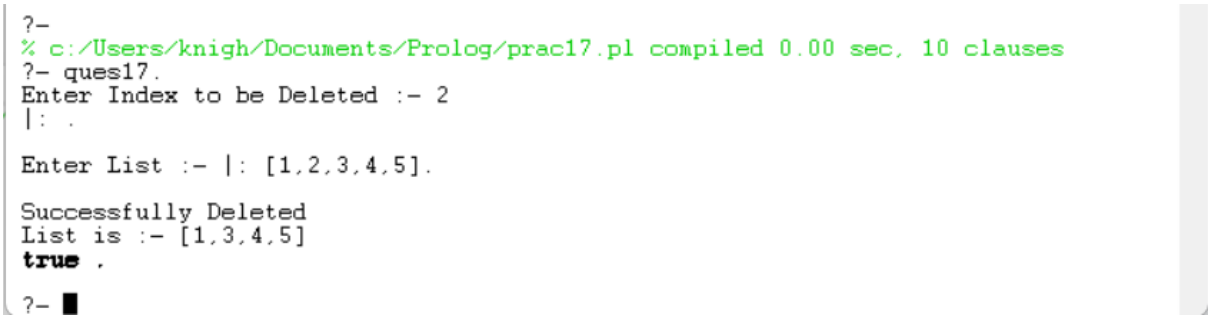
Ans:

```
insert_nth(I, 1, L, [I|L]).
insert_nth(I, N, [H|T], [H|R]) :-
    N > 1,
    N1 is N-1,
    insert_nth(I, N1, T, R).
take_input(List) :-
    write('Enter a list: '),
    read(List).
take_input_element(E) :-
    write('Enter the element to insert: '),
    read(E).
take_input_position(P) :-
    write('Enter the position to insert: '),
    read(P).
insert :-
    take_input(List),
    take_input_element(E),
    take_input_position(P),
    insert_nth(E, P, List, Result),
    write('Original list: '),
    write(List),
    nl,
    write('Element inserted list: '),
    write(Result).
```



Q17. 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.
Ans:

```
show([_H|T]):- nl,write("List is :- "),write(T).

len([],0).
len([_H|T],A):-len(T,A1),A is A1+1.
removelast([H|T]):- reverse([H|T],[_H1|T1]),reverse(T1,[_X|Y]),write("List is :- "),write(Y).
delete_nth(_N,[]):-write(" Nothing to Delete : List is Empty (UnderFlow).").
delete_nth(N,[H|T]):- len([H|T],L),delete_nth(N,[H|T],1,[_],L).
delete_nth(N,[_H|T],C,[H1|T1],_L):- C=:=N,append([H1|T1],T,V),write("Successfully
Deleted"),show(V).
delete_nth(N,[H|T],C,[H1|T1],L):- C<N,S is C+1,append([H1|T1],[H],V),delete_nth(N,T,S,V,L).
delete_nth(N,[H|T],_C,[_H1|_T1],L):- N=:=L,removelast([H|T]).
ques17:- write("Enter Index to be Deleted :- "),read(P),nl,write("Enter List :-
"),read(L),nl,delete_nth(P,L).
```



Q18. Write a Prolog Program to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.

Ans:

```
merge(L1,L2,L3).
merge([Head|Tail], L1, [Head|Merged]) :-
    merge(Tail, L2, L3).

input_list(List) :-
    write('Enter a list (e.g., [1, 2, 3]): '),
    read(List).

check_merge_lists :-
    write('Enter the first list: '),
    input_list(List1),
    write('Enter the second list: '),
    input_list(List2),
    merge(List1, List2, Merged),
    write('Merged list: '),
    write(Merged).
```