SUBMITTED TO : MS. NEHA MA'AM

NAME : GARIMA ROHILLA

COURSE : B. SC. (H) COMPUTER SCIENCE , III YEAR , VI SEMESTER

COLLEGE ROLL NO : CSC/21/37

UNIVERSITY ROLL NO : 21059570010

PRACTICAL FILE FOR CORE PAPER XIII : ARTIFICIAL INTELLIGENCE

6. Write a Prolog program to implement power (Num,Pow, Ans) : where Num is raised to the power Pow to get Ans.

EDITOR CODE :

```
:- initialization(main).

% base case
power(_, 0, 1).

% recursive case
power(Num, Pow, Ans) :-
    Pow > 0,
    NewPow is Pow - 1,
    power(Num, NewPow, NewAns),
    Ans is Num * NewAns.

% main predicate to read input and compute power
main :-
    write('Enter the base number: '),
    read(Num),
    write('Enter the power: '),
    read(Pow),
    power(Num, Pow, Ans),
    write('The result of '), write(Num), write(' raised to the power '),
write(Pow), write(' is: '), write(Ans), nl.
```

Alq2.pl | Alq3.pl | Alq4.pl | Alq5.pl | Alq6.pl | Alq7.pl | Alq8.pl | Alq9.pl | Alq

Alq6.pl

```
1   :- initialization(main).
2
3   % base case
4   power(_, 0, 1).
5
6   % recursive case
7   power(Num, Pow, Ans) :-
8       Pow > 0,
9       NewPow is Pow - 1,
10      power(Num, NewPow, NewAns),
11      Ans is Num * NewAns.
12
13  % main predicate to read input and compute power
14  main :-
15      write('Enter the base number: '),
16      read(Num),
17      write('Enter the power: '),
18      read(Pow),
19      power(Num, Pow, Ans),
20      write('The result of '), write(Num), write(' raised to the power '), write(Pow), write(' is: '), write(Ans), nl.
21
```

OUTPUT :

SWI-Prolog (AMD64, Multi-threaded, version 9.2.1)

File Edit Settings Run Debug Help

```
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.1)
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).

?-
% c:/Users/HP/Desktop/ai programs/Alq6.pl compiled 0.00 sec, 4 clauses
Enter the base number: 6.
Enter the power: |    5.
The result of 6 raised to the power 5 is: 7776
```

7. Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes then numbers to be multiplied and R represents the result.

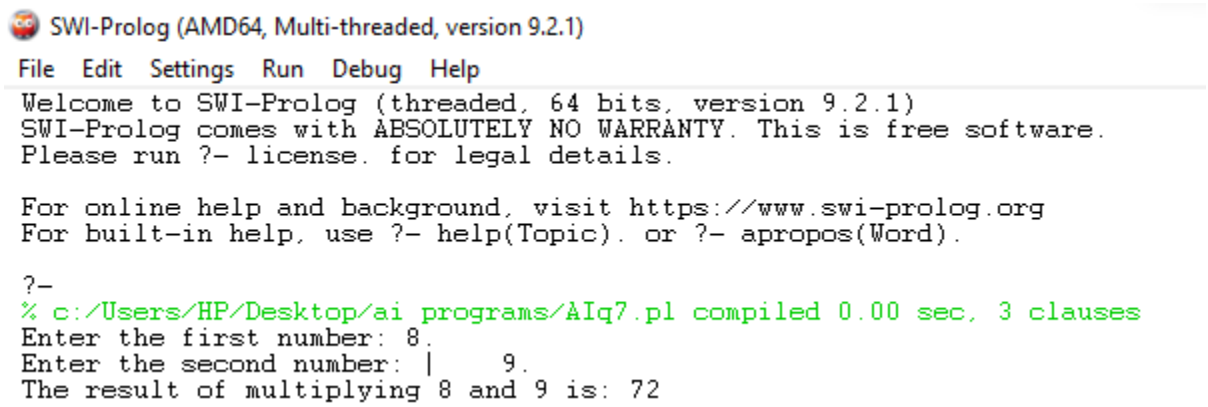
EDITOR CODE :

```
:- initialization(main).

multi(N1, N2, R) :-
    R is N1 * N2.

% main predicate to read input and compute multiplication
main :-
    write('Enter the first number: '),
    read(N1),
    write('Enter the second number: '),
    read(N2),
    multi(N1, N2, R),
    write('The result of multiplying '), write(N1), write(' and '), write(N2),
write(' is: '), write(R), nl.
```

OUTPUT :


SWI-Prolog (AMD64, Multi-threaded, version 9.2.1)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.1)
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).
?-
% c:/Users/HP/Desktop/ai programs/AIq7.pl compiled 0.00 sec, 3 clauses
Enter the first number: 8.
Enter the second number: | 9.
The result of multiplying 8 and 9 is: 72

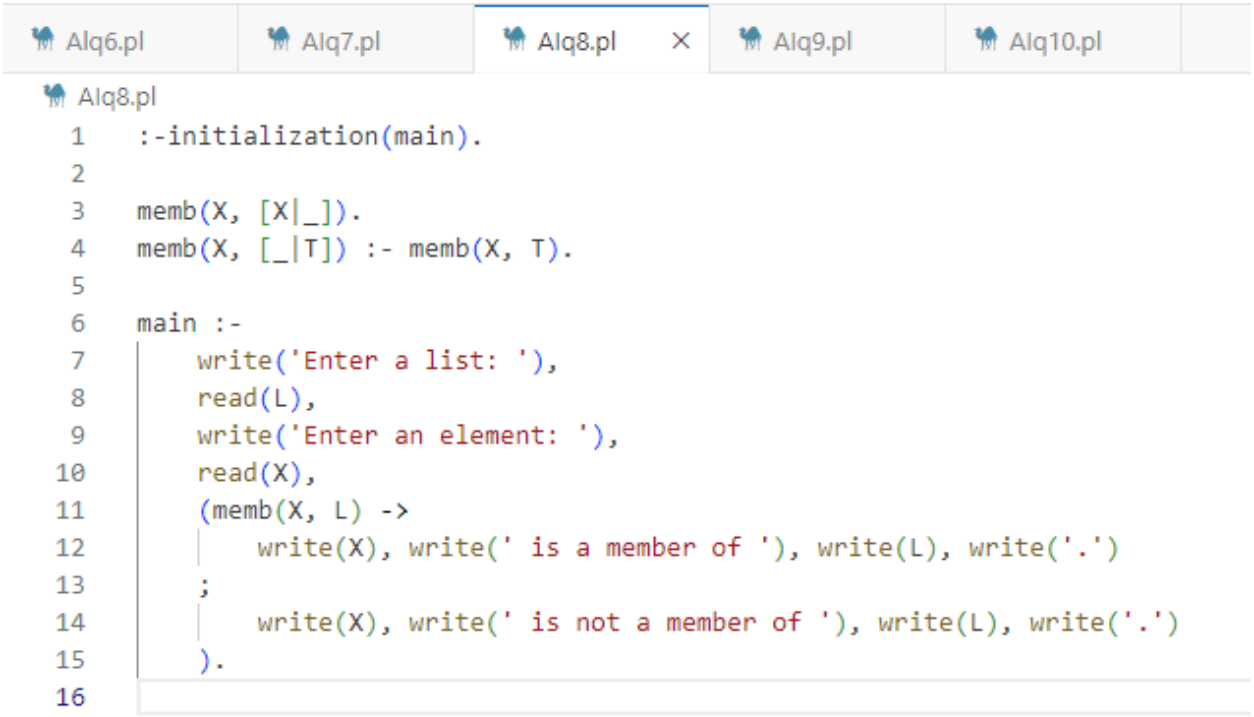
8. Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.

EDITOR CODE :

```
:-initialization(main).

memb(X, [X|_]).
memb(X, [_|T]) :- memb(X, T).

main :-
    write('Enter a list: '),
    read(L),
    write('Enter an element: '),
    read(X),
    (memb(X, L) ->
        write(X), write(' is a member of '), write(L), write('.')
    ;
        write(X), write(' is not a member of '), write(L), write('.')
    ).
```



OUTPUT :


SWI-Prolog (AMD64, Multi-threaded, version 9.2.1)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.1)
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).
?-
% c:/Users/HP/Desktop/ai programs/AIq8.pl compiled 0.00 sec, 4 clauses
Enter a list: [3,4,5,6,7,8].
Enter an element: | 8.
8 is a member of [3,4,5,6,7,8].

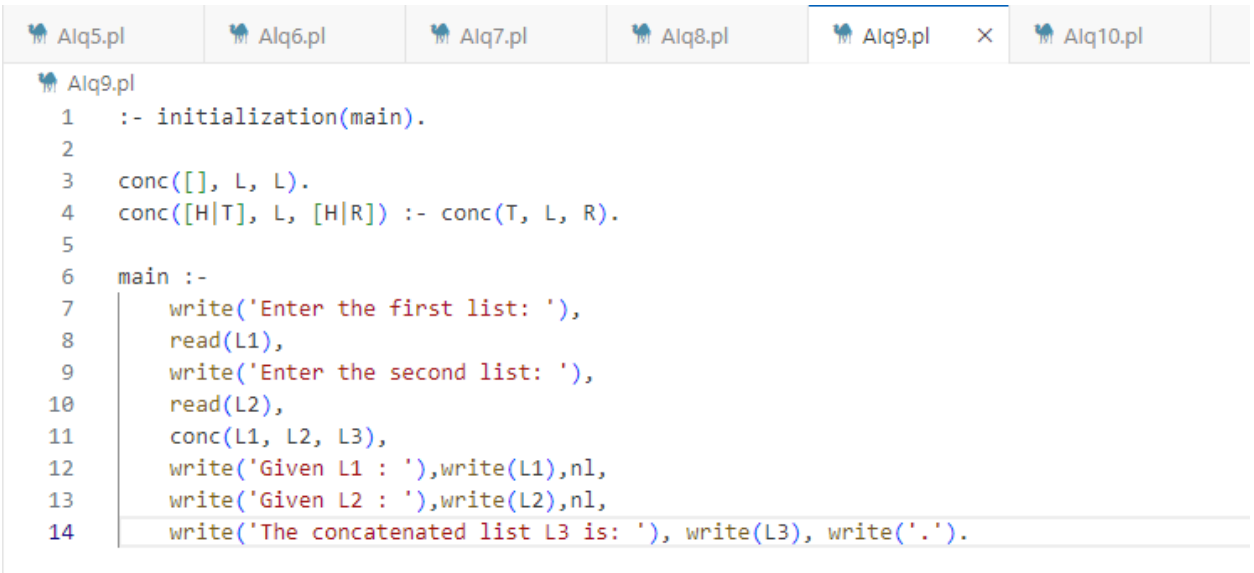
9. Write a Prolog program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.

EDITOR CODE :

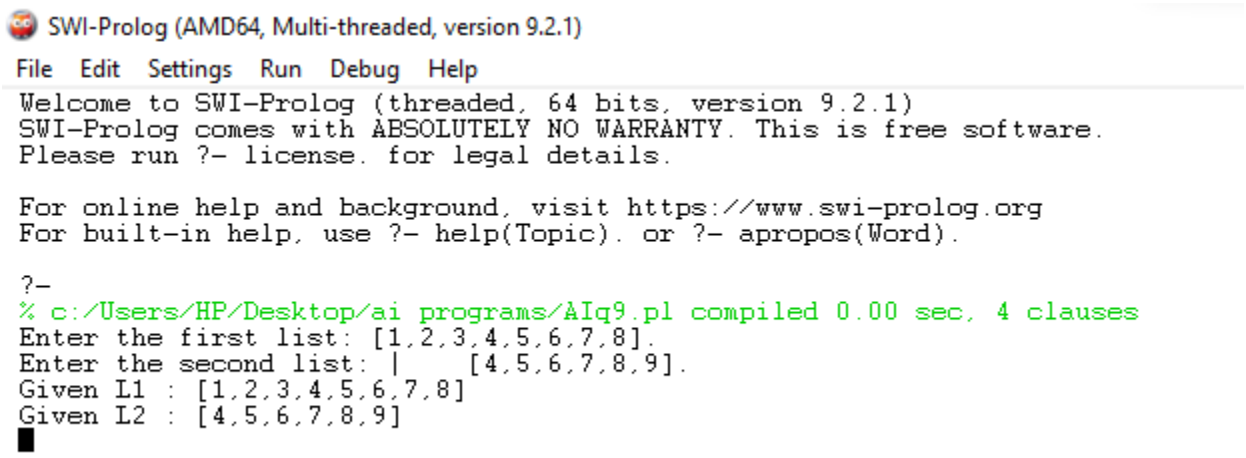
```
:- initialization(main).

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

main :-
    write('Enter the first list: '),
    read(L1),
    write('Enter the second list: '),
    read(L2),
    conc(L1, L2, L3),
    write('Given L1 : '),write(L1),nl,
    write('Given L2 : '),write(L2),nl,
    write('The concatenated list L3 is: '), write(L3), write('.').
```



OUTPUT :


SWI-Prolog (AMD64, Multi-threaded, version 9.2.1)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.1)
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).
?-
% c:/Users/HP/Desktop/ai programs/AIq9.pl compiled 0.00 sec, 4 clauses
Enter the first list: [1,2,3,4,5,6,7,8].
Enter the second list: | [4,5,6,7,8,9].
Given L1 : [1,2,3,4,5,6,7,8]
Given L2 : [4,5,6,7,8,9]

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

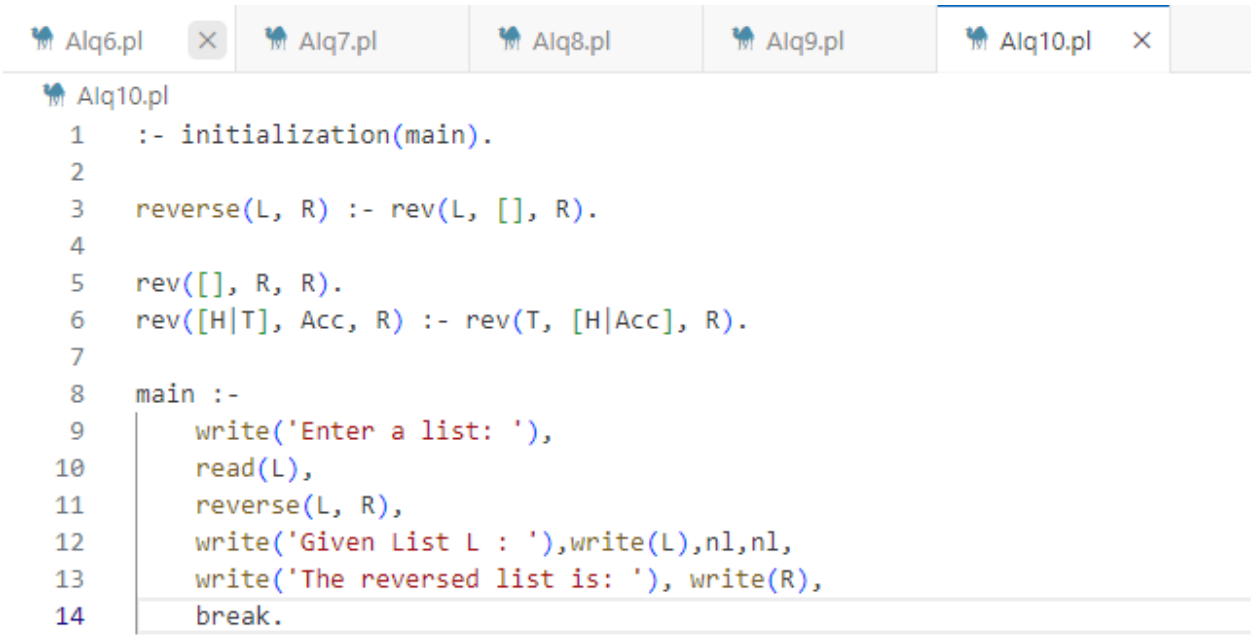
EDITOR CODE :

```
:- initialization(main).

reverse(L, R) :- rev(L, [], R).

rev([], R, R).
rev([H|T], Acc, R) :- rev(T, [H|Acc], R).

main :-
    write('Enter a list: '),
    read(L),
    reverse(L, R),
    write('Given List L : '),write(L),nl,nl,
    write('The reversed list is: '), write(R),
    break.
```



OUTPUT :

```
SWI-Prolog (AMD64, Multi-threaded, version 9.2.1)
File  Edit  Settings  Run  Debug  Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.2.1)
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).

?-
% c:/Users/HP/Desktop/ai programs/AIq10.pl compiled 0.00 sec, 5 clauses
Enter a list: [11,22,33,44,55,66].
Given List L : [11,22,33,44,55,66]

The reversed list is: [66,55,44,33,22,11]
% Break level 1
[1]  ?-
```