

ARTIFICIAL INTELLIGENCE PRACTICAL FILE

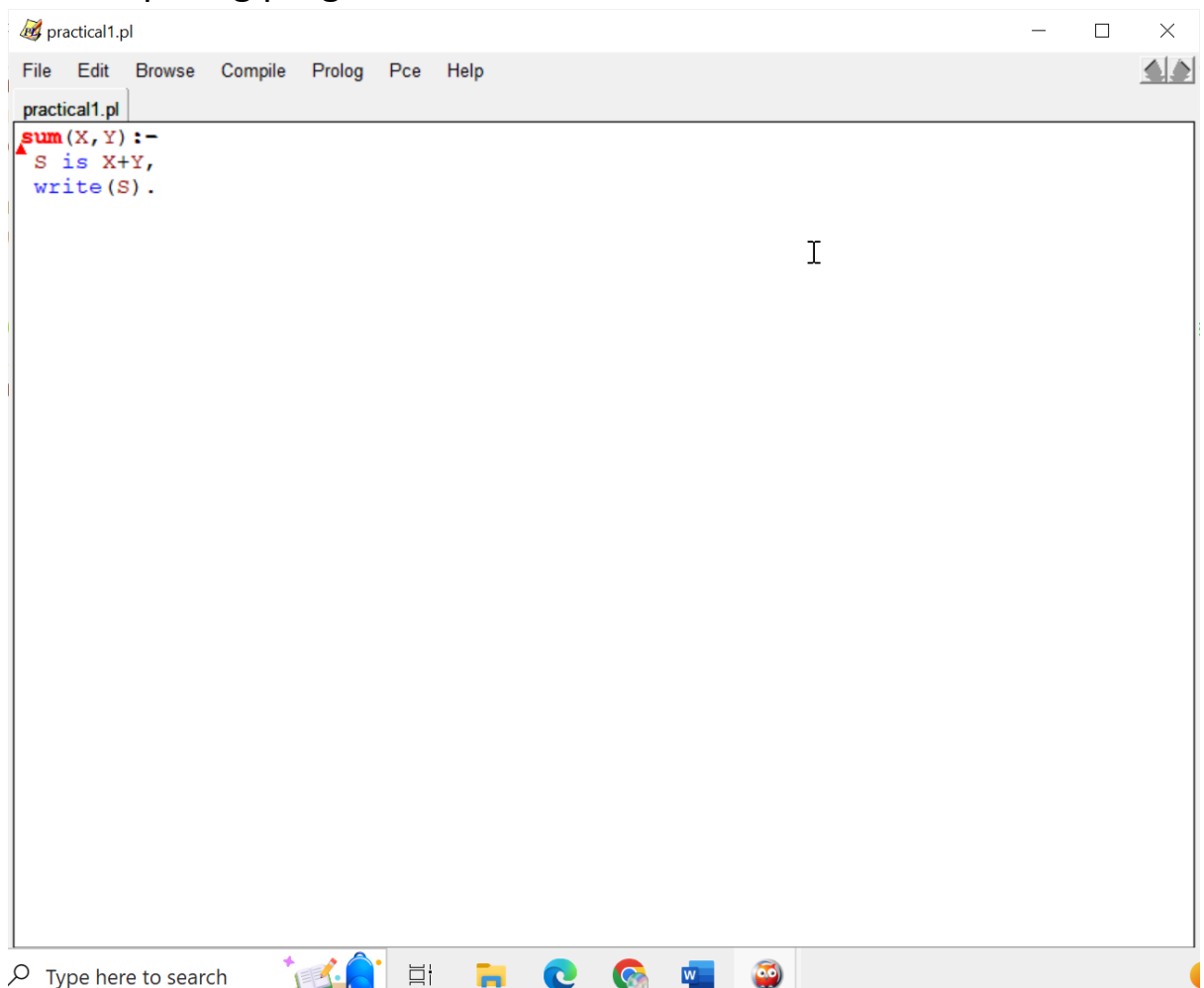
AADITYA KEDIYAL

BSc(Hons) Computer Science

20201401

Ramanujan College

1. Write a prolog program to calculate the sum of two numbers



The screenshot shows a Prolog IDE window titled "practical1.pl". The menu bar includes "File", "Edit", "Browse", "Compile", "Prolog", "Pce", and "Help". The code editor contains the following Prolog program:

```
sum(X,Y):-  
    S is X+Y,  
    write(S).
```

The cursor is positioned at the end of the second line of code. The Windows taskbar is visible at the bottom, showing the search bar and several application icons.

OUTPUT

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical1.pl compiled 0.00 sec
, 1 clauses
?- sum(11,12).
23
true.
?-
```

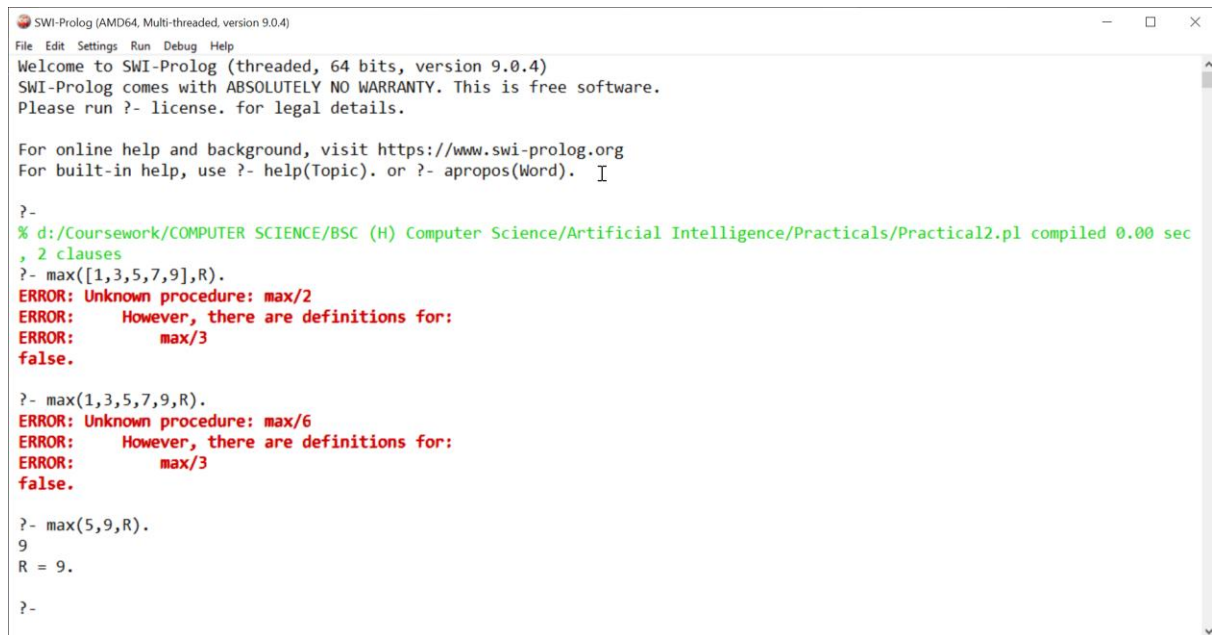
2. Write a Prolog program to implement $\text{max}(X, Y, M)$ so that M is the maximum of two numbers X and Y .

```
practical2.pl
File Edit Browse Compile Prolog Pce Help
practical2.pl
/* Max of two #.s */

/* without list. */
max(X,Y,R):-
    X>=Y ->
        R is X,
        write(R)
    ;
    R is Y,
    write(R) .

/* with list. */
grandiose([H|T],R):-
    H>T ->
        R is H,
        write(R)
    ;
    R is T,
    write(T) .
```

OUTPUT



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

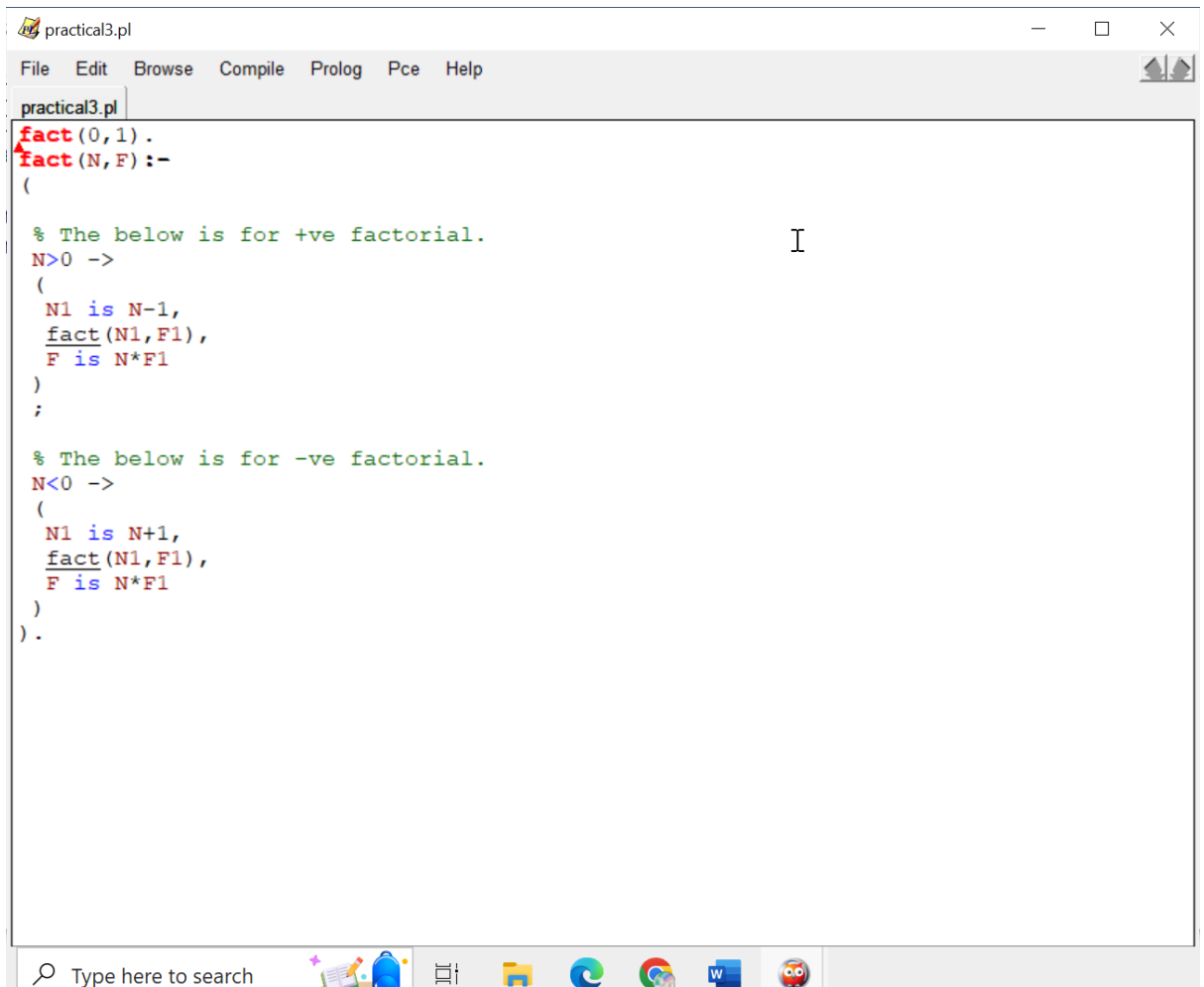
?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical2.pl compiled 0.00 sec
, 2 clauses
?- max([1,3,5,7,9],R).
ERROR: Unknown procedure: max/2
ERROR:         However, there are definitions for:
ERROR:         max/3
false.

?- max(1,3,5,7,9,R).
ERROR: Unknown procedure: max/6
ERROR:         However, there are definitions for:
ERROR:         max/3
false.

?- max(5,9,R).
9
R = 9.

?-
```

3. Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.

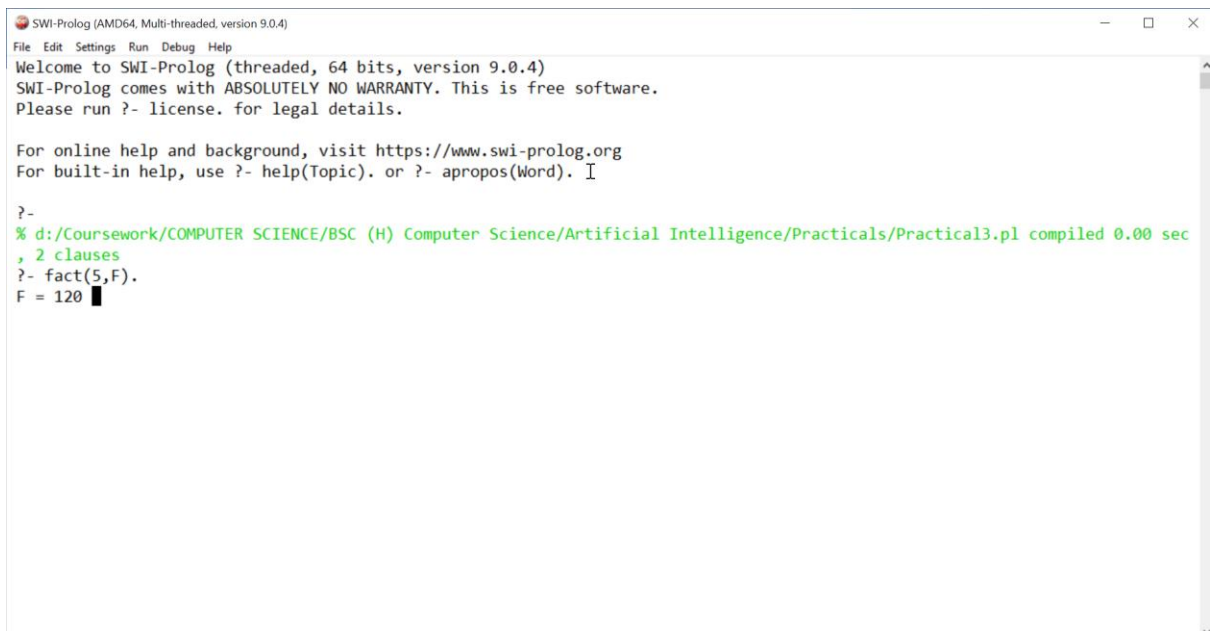


```
practical3.pl
fact(0,1).
fact(N,F):-
(
    % The below is for +ve factorial.
    N>0 ->
    (
        N1 is N-1,
        fact(N1,F1),
        F is N*F1
    )
;

    % The below is for -ve factorial.
    N<0 ->
    (
        N1 is N+1,
        fact(N1,F1),
        F is N*F1
    )
).

Type here to search
```

OUTPUT

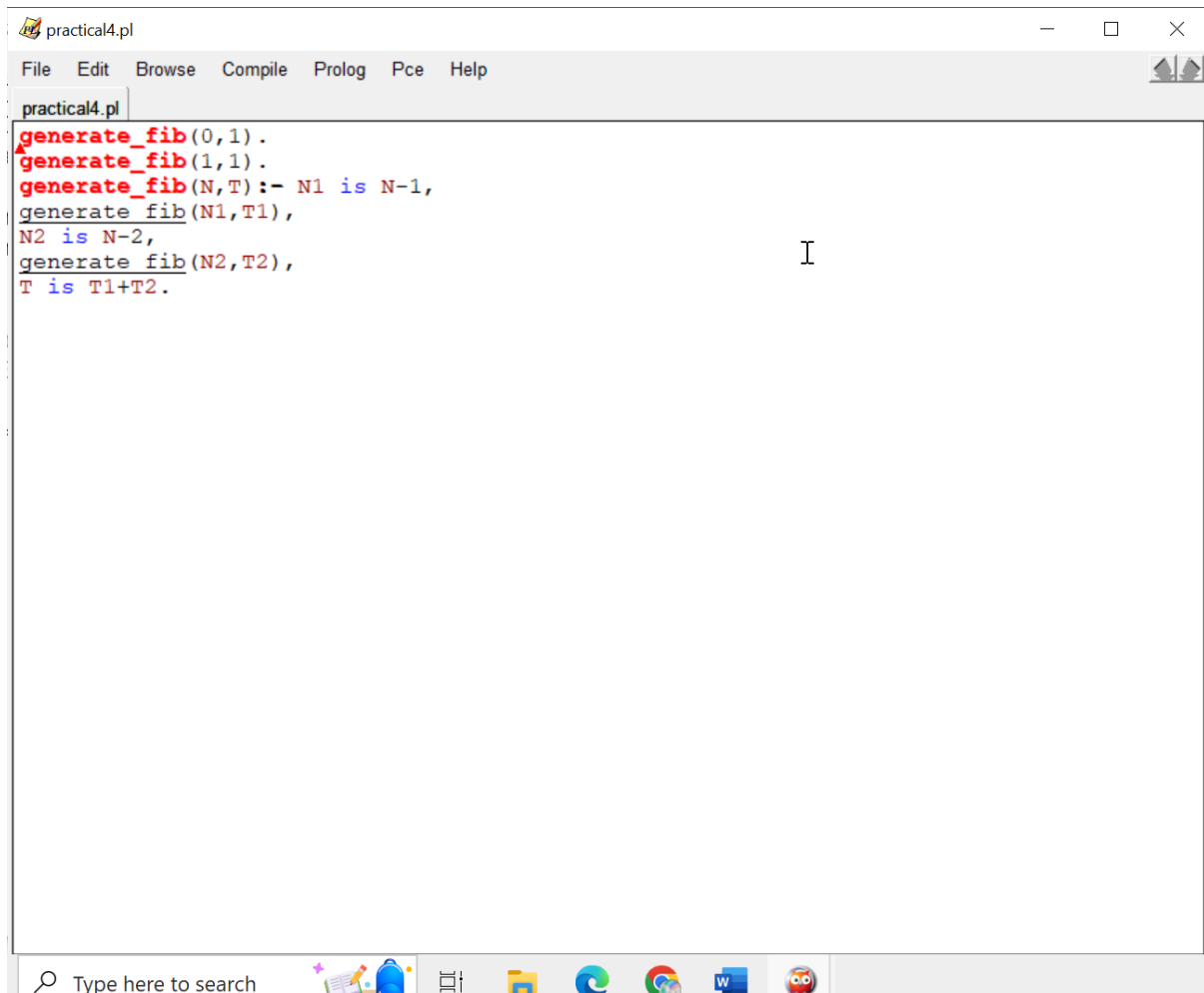


```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

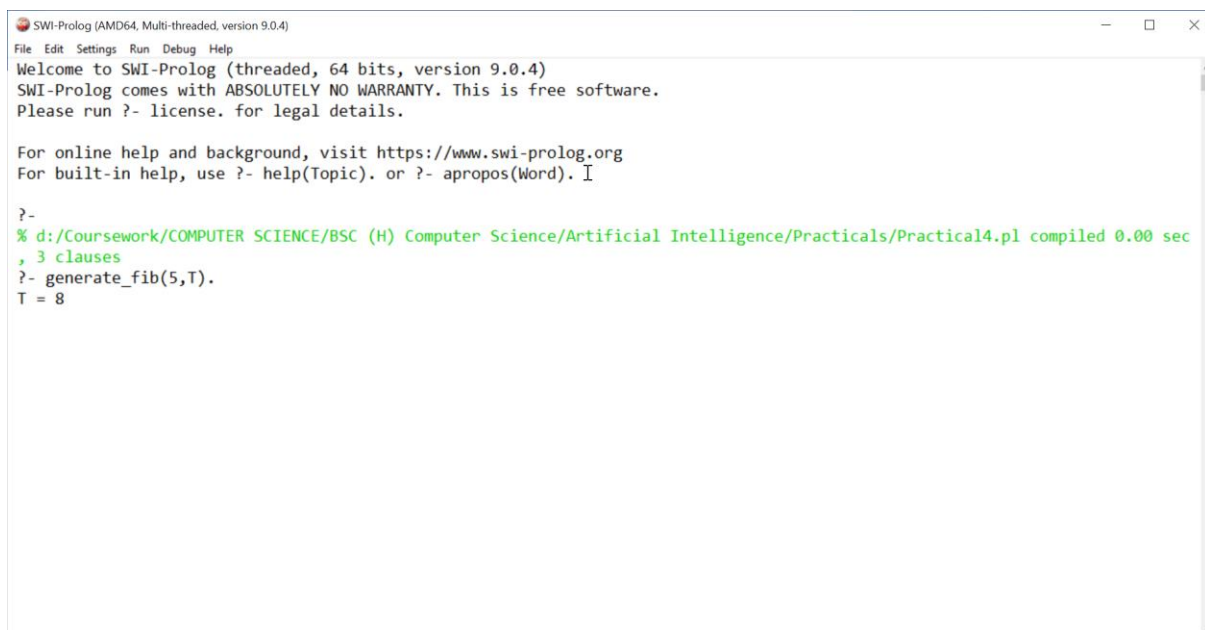
?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical3.pl compiled 0.00 sec
, 2 clauses
?- fact(5,F).
F = 120
```

4. Write a program in PROLOG to implement generate_fib(N,T) where T represents the Nth term of the fibonacci series



```
practical4.pl
File Edit Browse Compile Prolog Pce Help
practical4.pl
generate_fib(0,1).
generate_fib(1,1).
generate_fib(N,T):- N1 is N-1,
generate_fib(N1,T1),
N2 is N-2,
generate_fib(N2,T2),
T is T1+T2.
```

OUTPUT

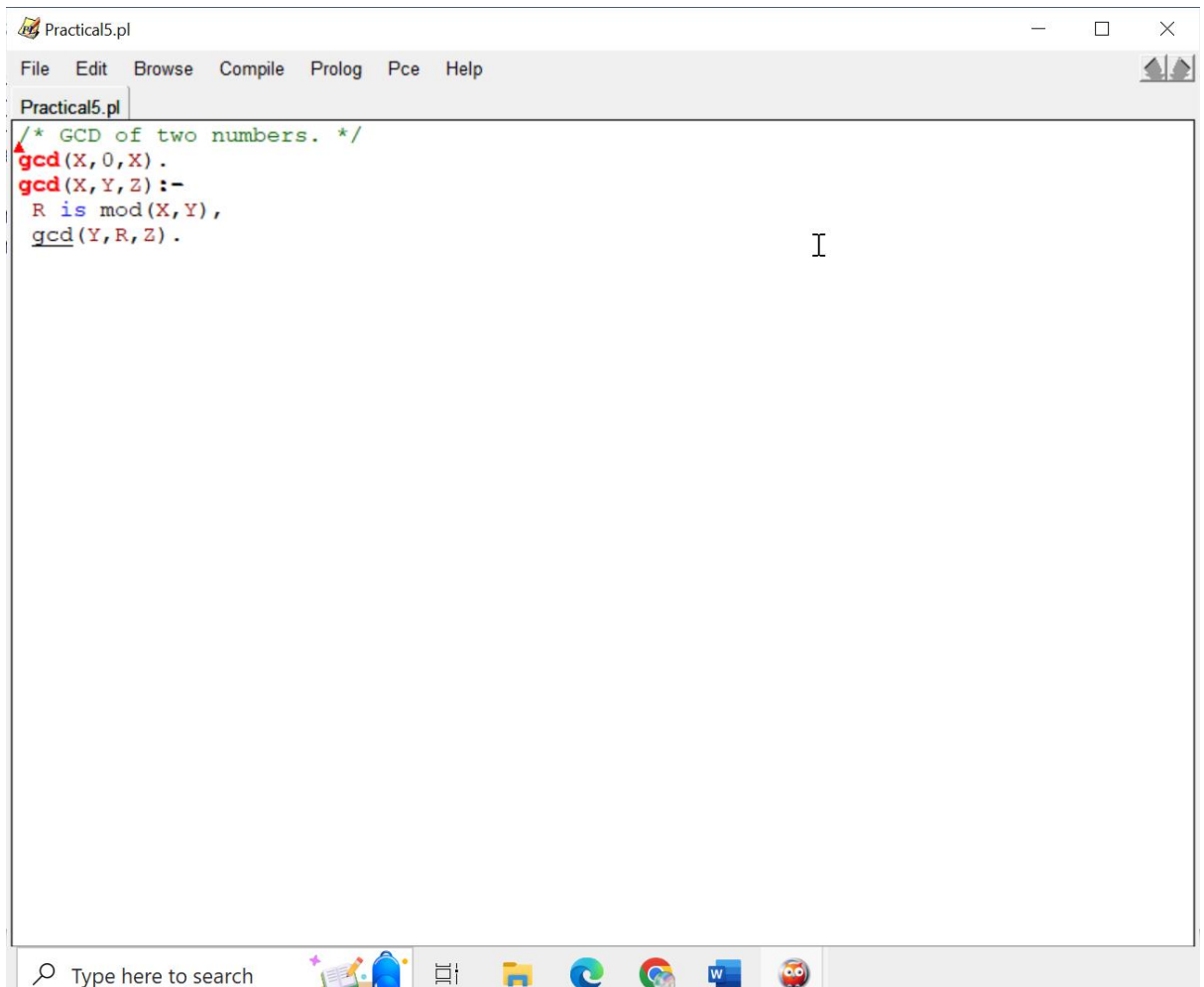


```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical4.pl compiled 0.00 sec
, 3 clauses
?- generate_fib(5,T).
T = 8
```

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



```
Practical5.pl
File Edit Browse Compile Prolog Pce Help
Practical5.pl
/* GCD of two numbers. */
gcd(X,0,X).
gcd(X,Y,Z):-
    R is mod(X,Y),
    gcd(Y,R,Z).
```

OUTPUT

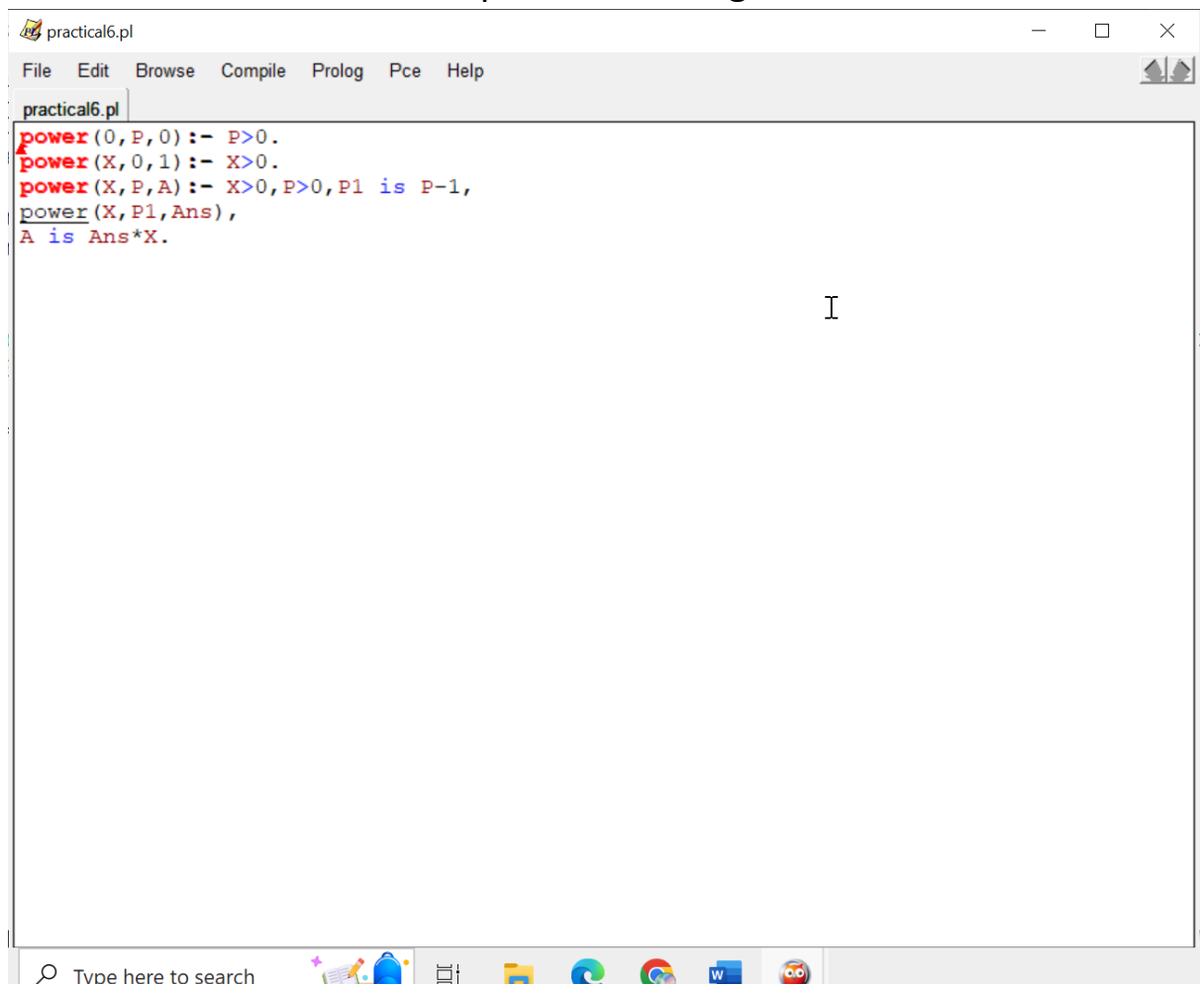


```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical5.pl compiled 0.00 sec
, 2 clauses
?- gcd(18,3,X).
X = 3
```

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



```
practical6.pl
File Edit Browse Compile Prolog Pce Help
practical6.pl
power(0,P,0):- P>0.
power(X,0,1):- X>0.
power(X,P,A):- X>0,P>0,P1 is P-1,
power(X,P1,Ans),
A is Ans*X.
```

OUTPUT

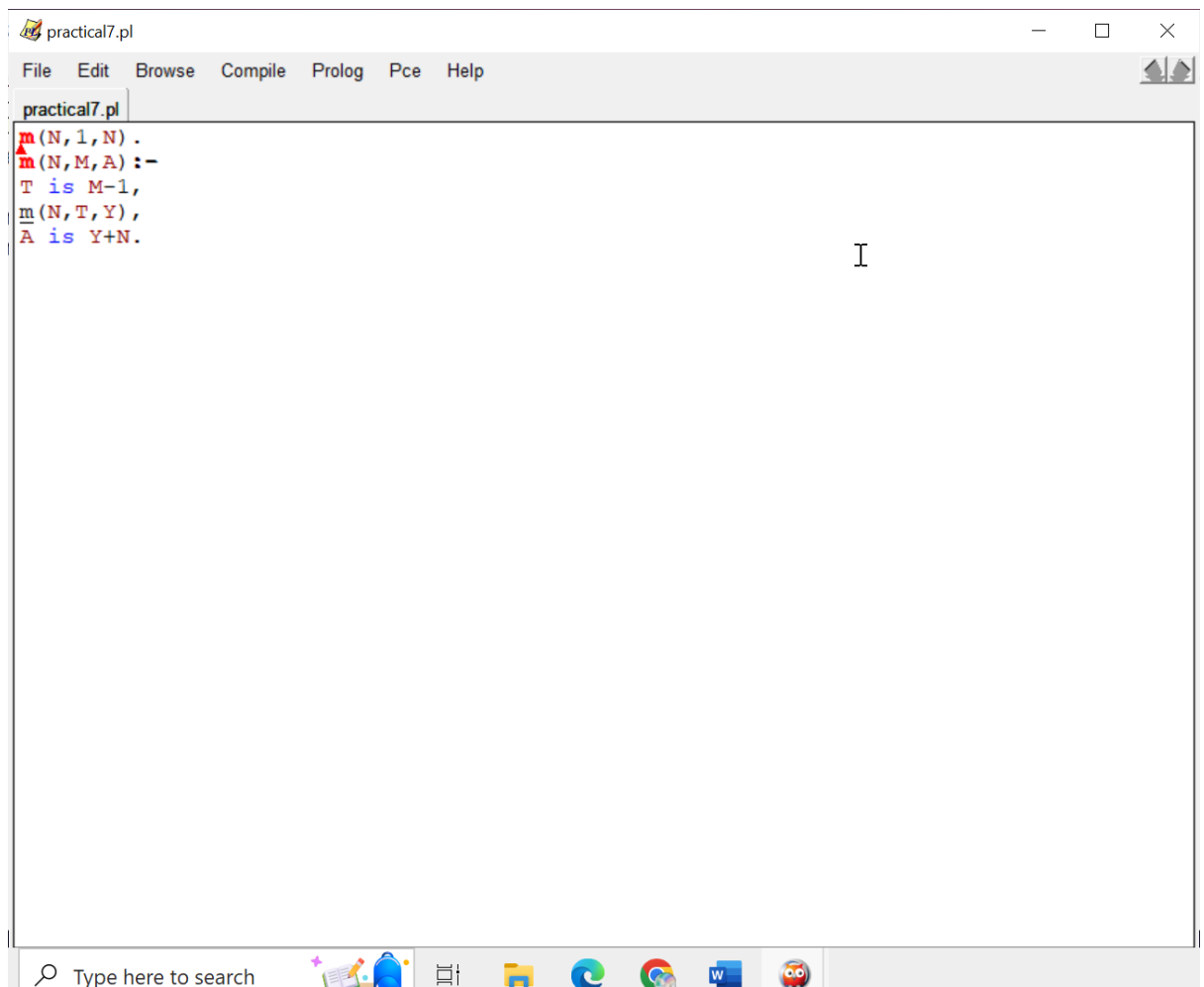


```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical6.pl compiled 0.00 sec
, 3 clauses
?- power(3,5,X).
X = 243
```

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



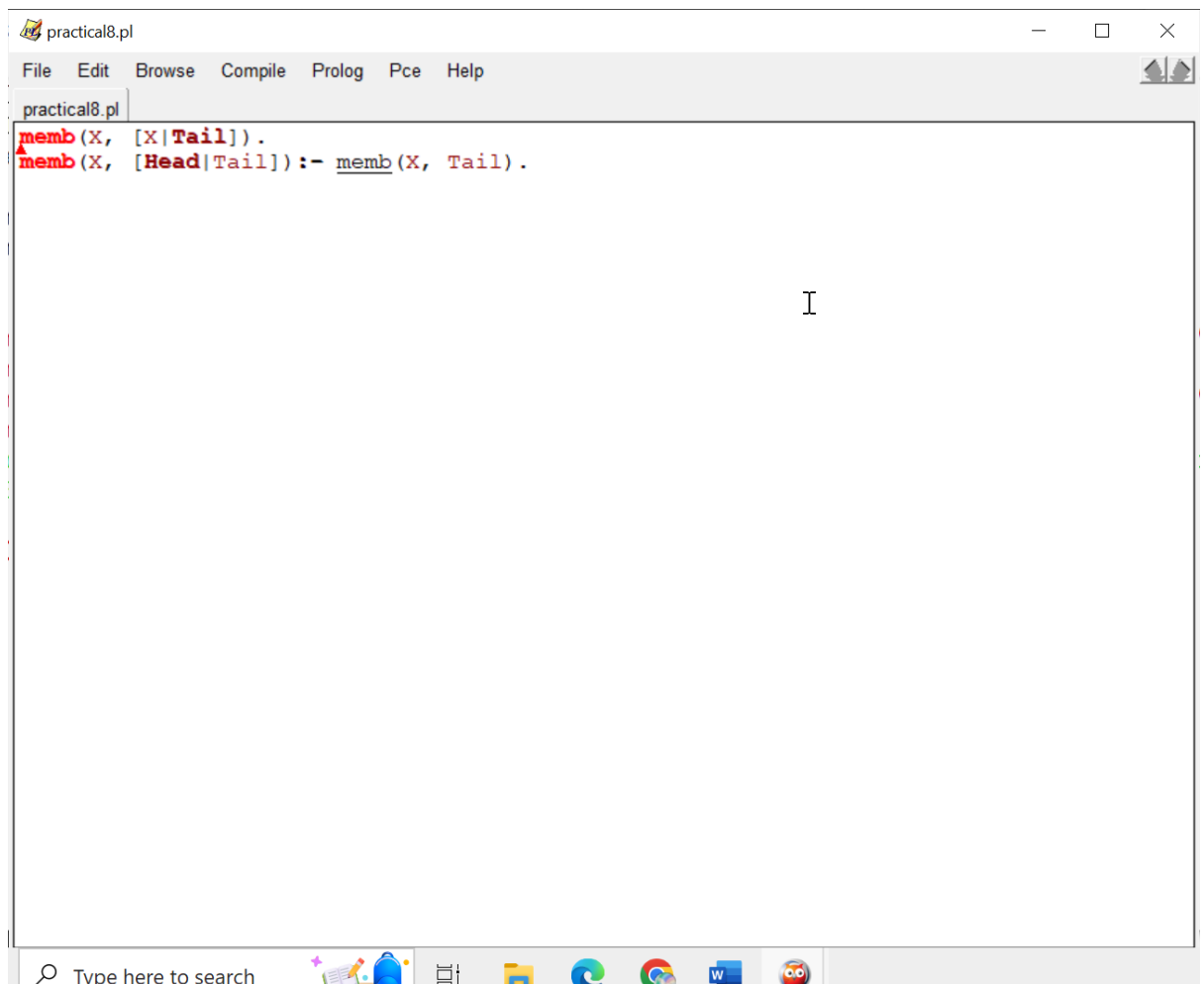
```
m(N, 1, N) .  
m(N, M, A) :-  
    T is M-1,  
    m(N, T, Y),  
    A is Y+N.
```

OUTPUT



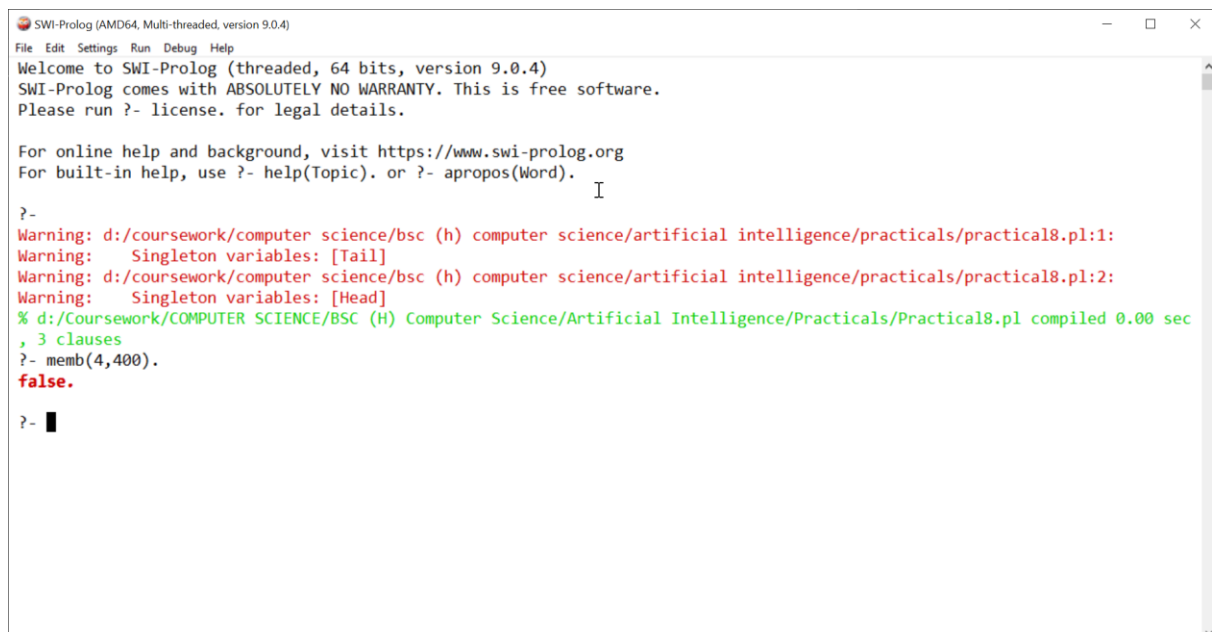
```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)  
File Edit Settings Run Debug Help  
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)  
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.  
Please run ?- license. for legal details.  
  
For online help and background, visit https://www.swi-prolog.org  
For built-in help, use ?- help(Topic). or ?- apropos(Word).  
  
?-  
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical7.pl compiled 0.00 sec  
, 2 clauses  
?- m(4,5,A).  
A = 20
```


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



```
practical8.pl
File Edit Browse Compile Prolog Pce Help
practical8.pl
memb(X, [X|Tail]).
memb(X, [Head|Tail]) :- memb(X, Tail).
```

OUTPUT



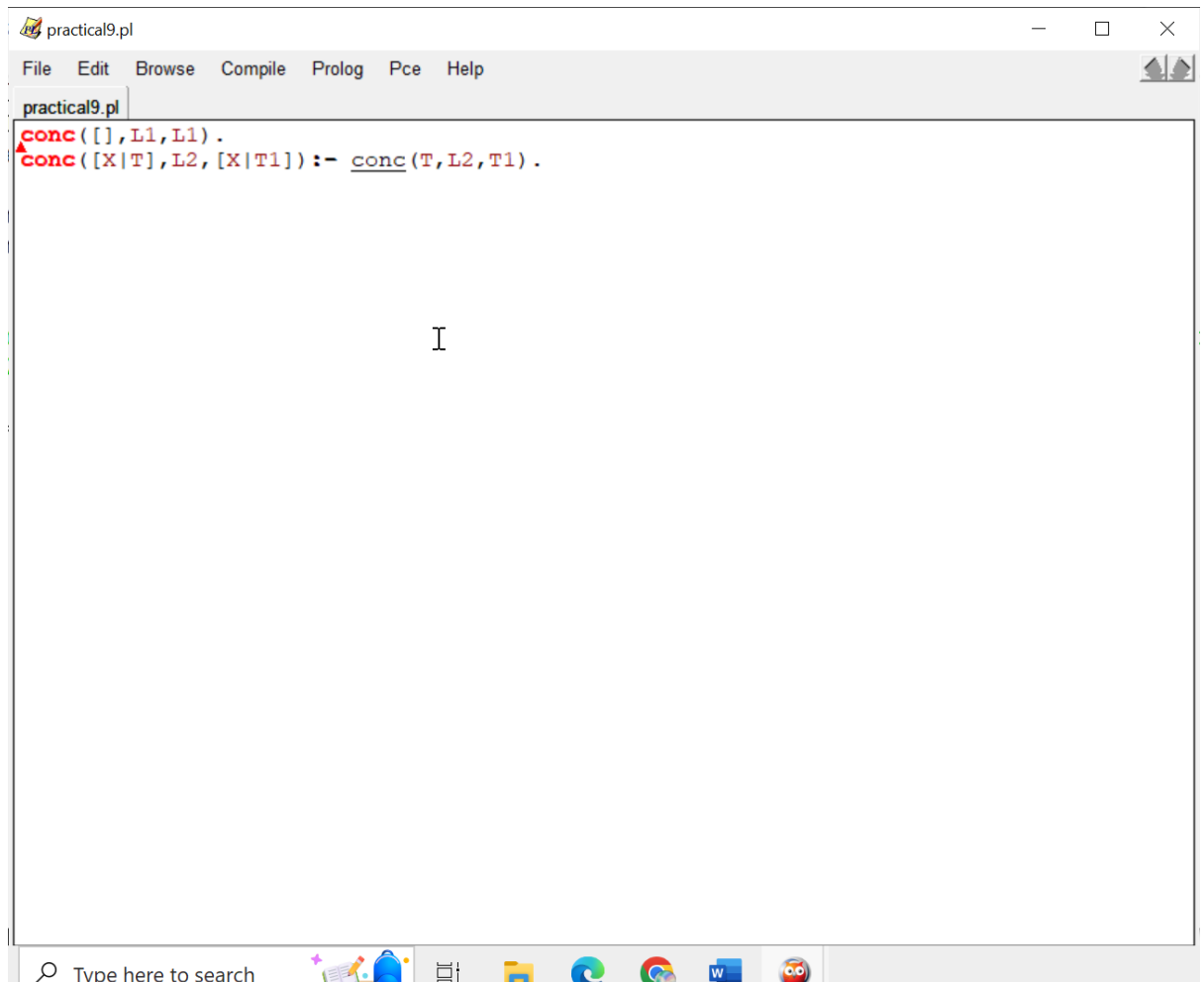
```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
Warning: d:/coursework/computer science/bsc (h) computer science/artificial intelligence/practicals/practical8.pl:1:
Warning: Singleton variables: [Tail]
Warning: d:/coursework/computer science/bsc (h) computer science/artificial intelligence/practicals/practical8.pl:2:
Warning: Singleton variables: [Head]
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical8.pl compiled 0.00 sec
, 3 clauses
?- memb(4,400).
false.

?-
```

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



```
practical9.pl
File Edit Browse Compile Prolog Pce Help
practical9.pl
conc([], L1, L1).
conc([X|T], L2, [X|T1]) :- conc(T, L2, T1).
```

OUTPUT



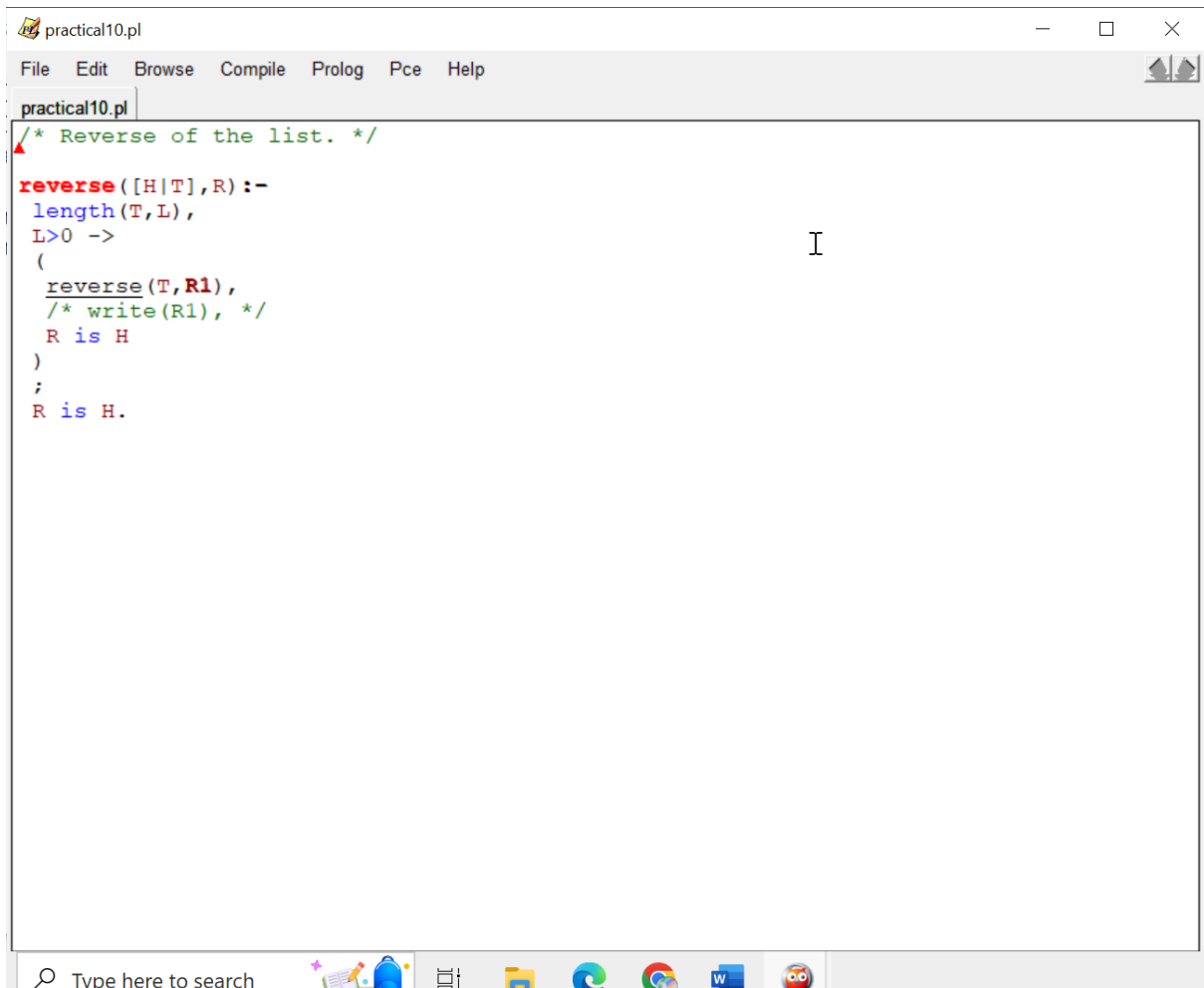
```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical9.pl compiled 0.00 sec
, 2 clauses
?- conc([1,3,5],[2,4,6],X).
X = [1, 3, 5, 2, 4, 6].

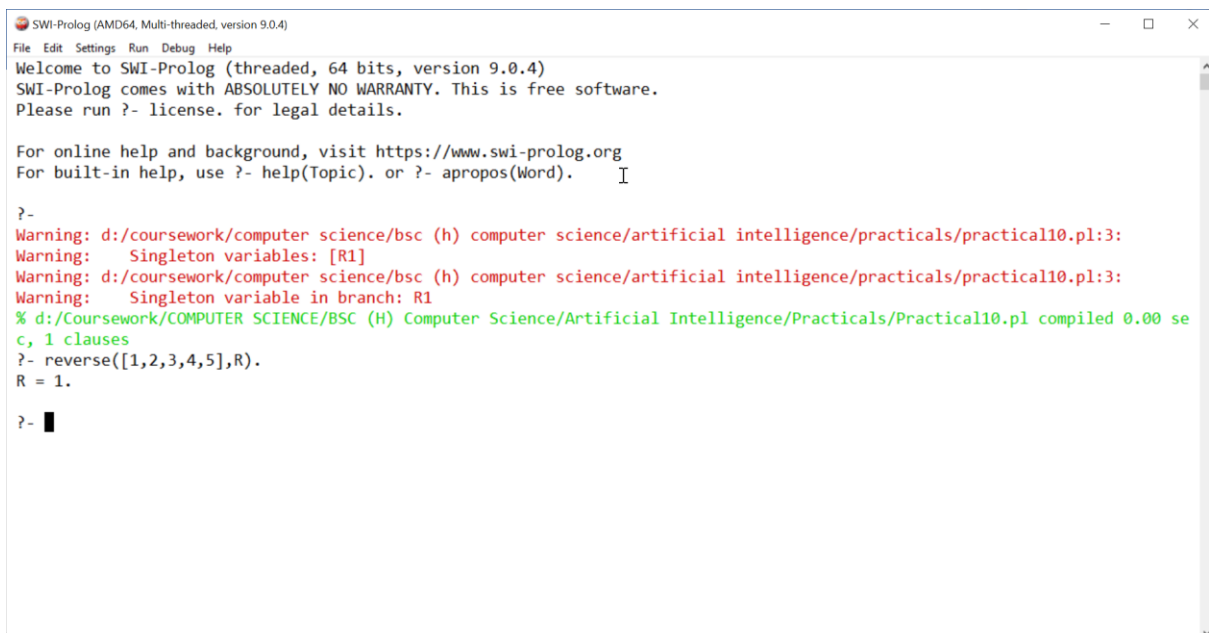
?-
```

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



```
practical10.pl
File Edit Browse Compile Prolog Pce Help
practical10.pl
/* Reverse of the list. */
reverse([H|T],R):-
    length(T,L),
    L>0 ->
    (
        reverse(T,R1),
        /* write(R1), */
        R is H
    )
    ;
    R is H.
```

OUTPUT



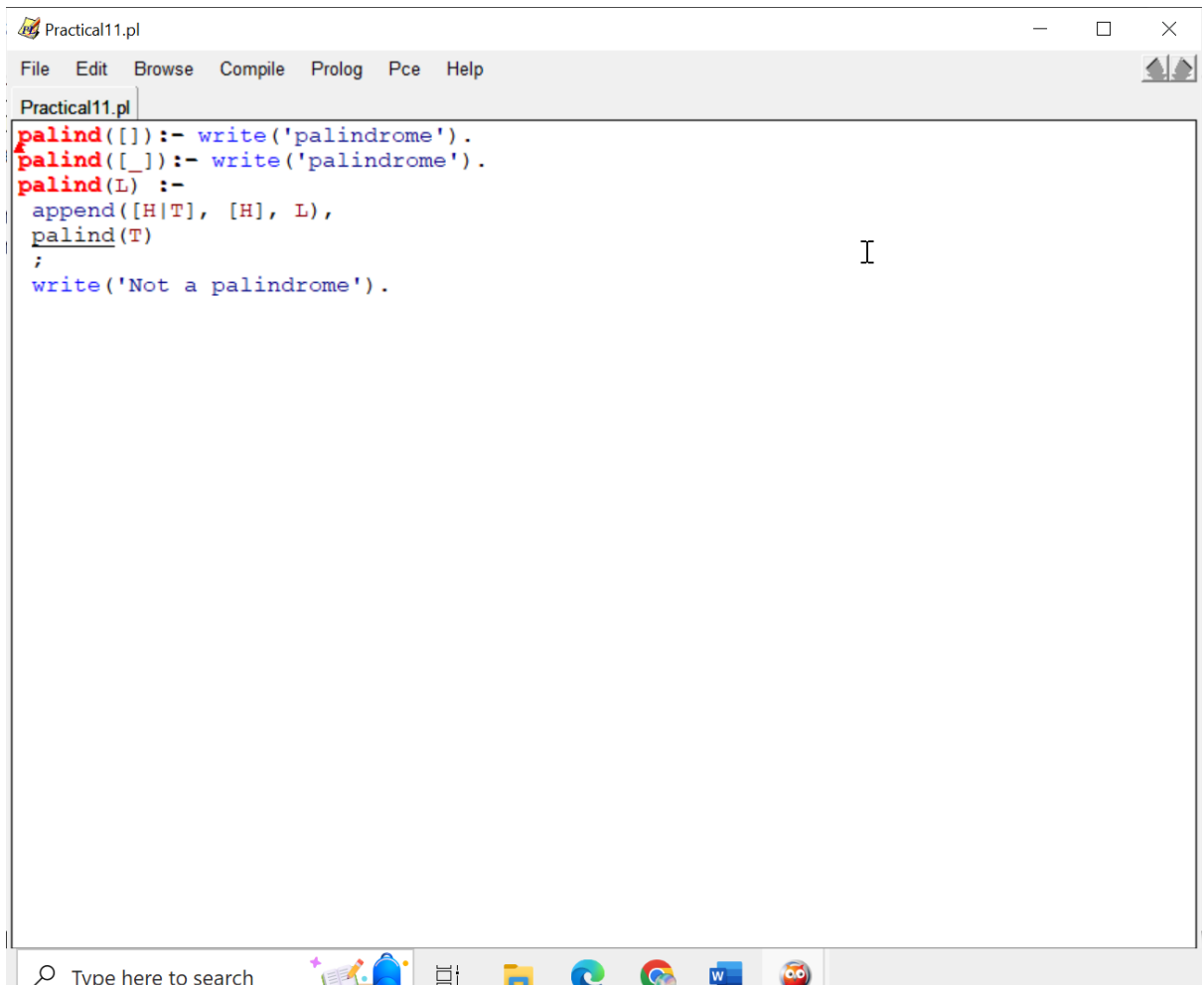
```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
Warning: d:/coursework/computer science/bsc (h) computer science/artificial intelligence/practicals/practical10.pl:3:
Warning: Singleton variables: [R1]
Warning: d:/coursework/computer science/bsc (h) computer science/artificial intelligence/practicals/practical10.pl:3:
Warning: Singleton variable in branch: R1
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical10.pl compiled 0.00 se
c, 1 clauses
?- reverse([1,2,3,4,5],R).
R = 1.

?-
```

11. Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not



```
Practical11.pl
File Edit Browse Compile Prolog Pce Help
Practical11.pl
palind([]):- write('palindrome').
palind([_]):- write('palindrome').
palind(L):-
    append([H|T], [H], L),
    palind(T)
;
write('Not a palindrome').
```

OUTPUT

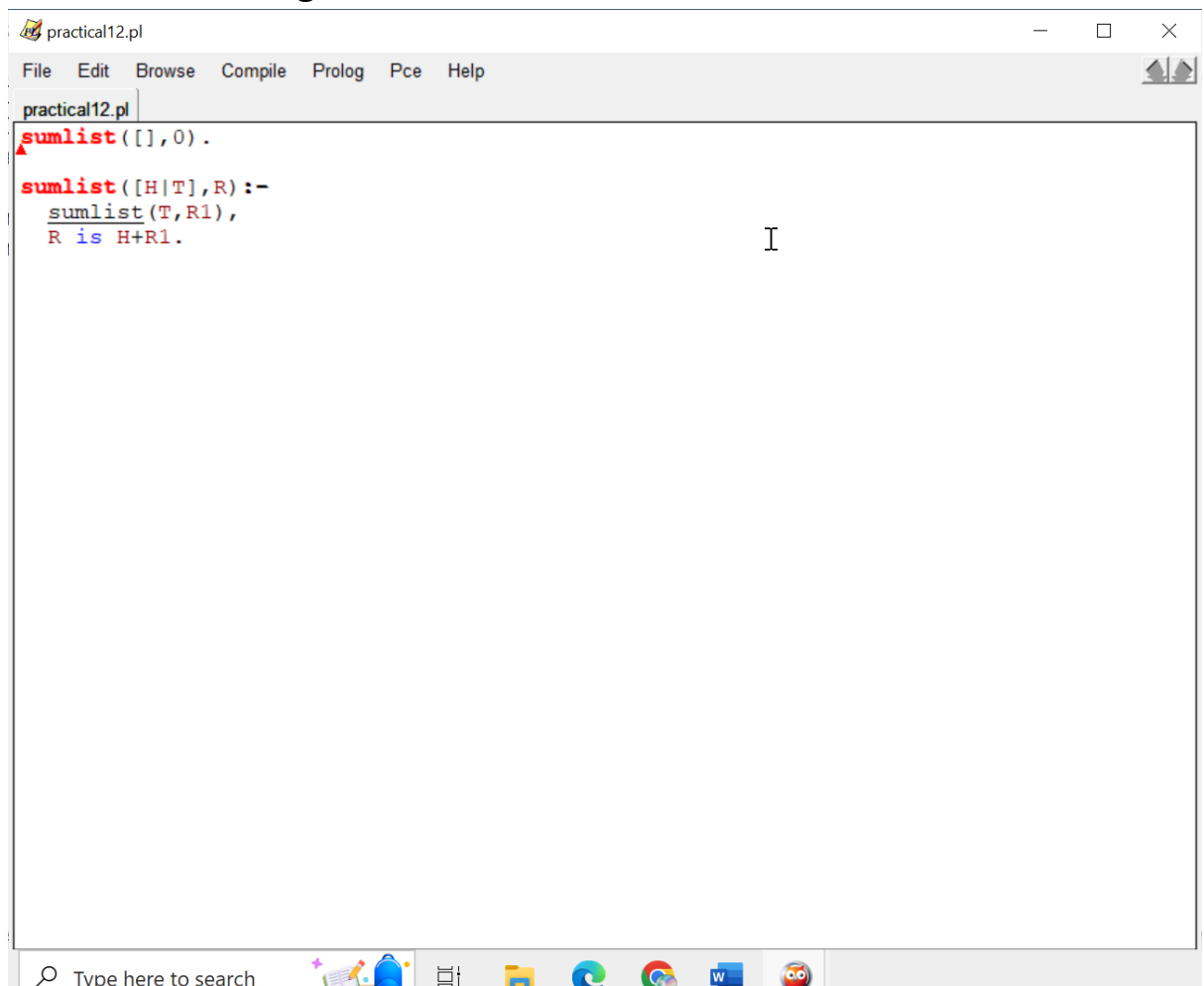


```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

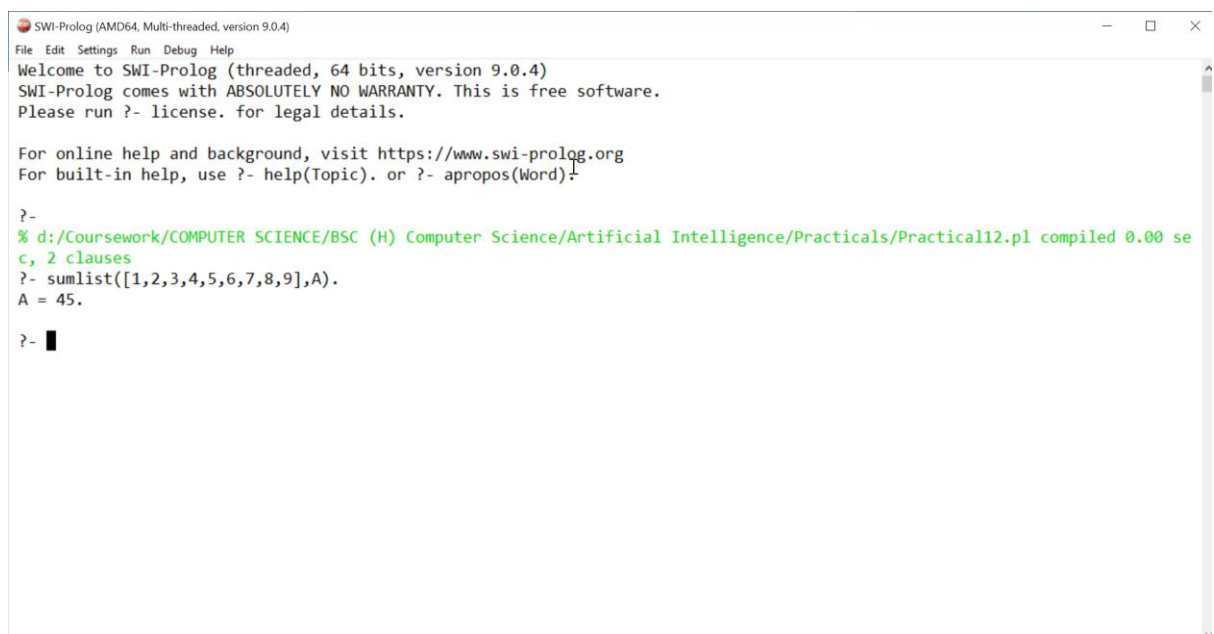
?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical11.pl compiled 0.00 se
c, 3 clauses
?- palind([n,i,t,i,n]).
palindrome
true
```

12. Write a Prolog program to implement `sumlist(L, S)` so that `S` is the sum of a given list `L`.



```
practical12.pl
sumlist([], 0).
sumlist([H|T], R) :-
    sumlist(T, R1),
    R is H+R1.
```

OUTPUT

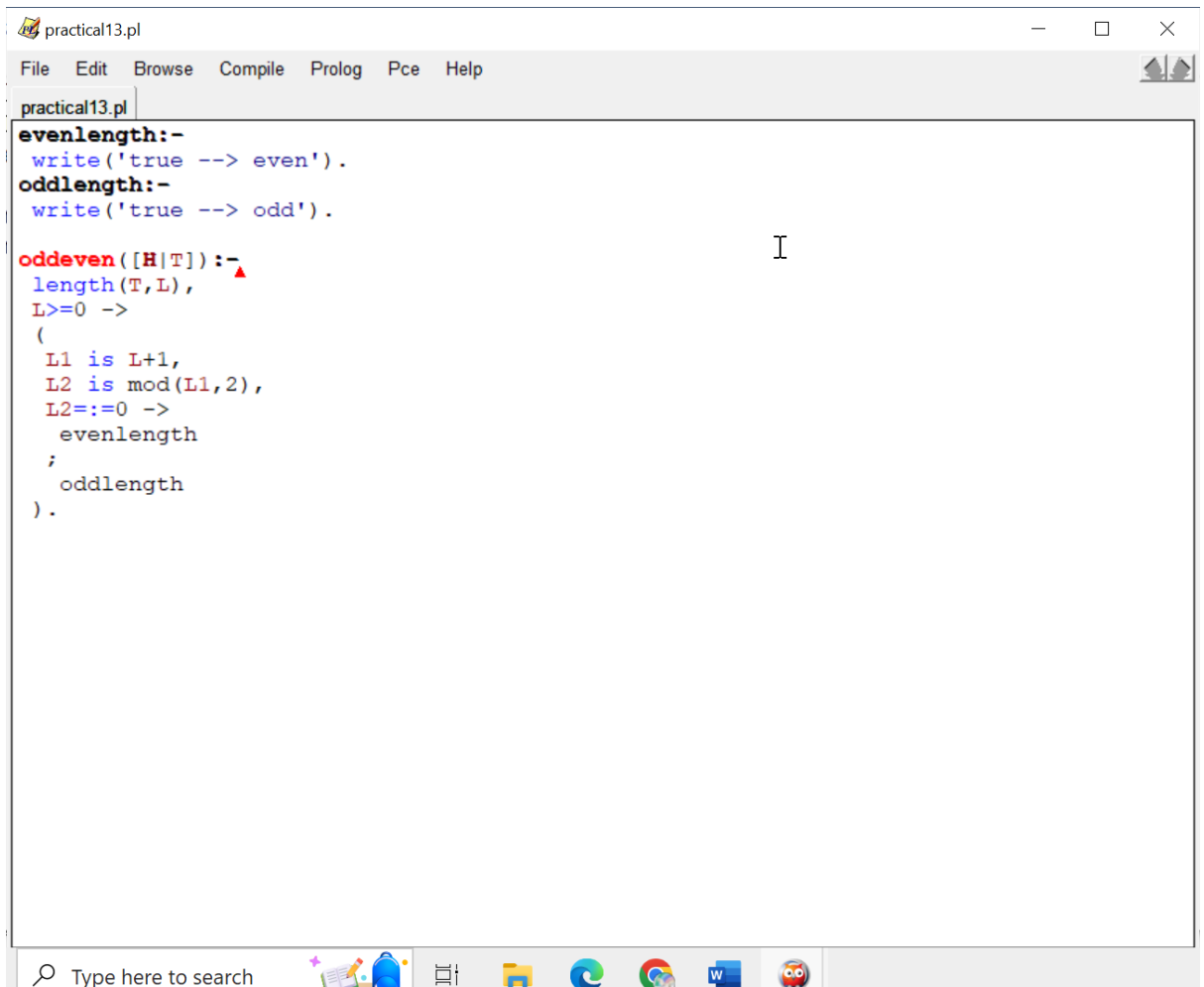


```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical12.pl compiled 0.00 se
c, 2 clauses
?- sumlist([1,2,3,4,5,6,7,8,9],A).
A = 45.
?-
```

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

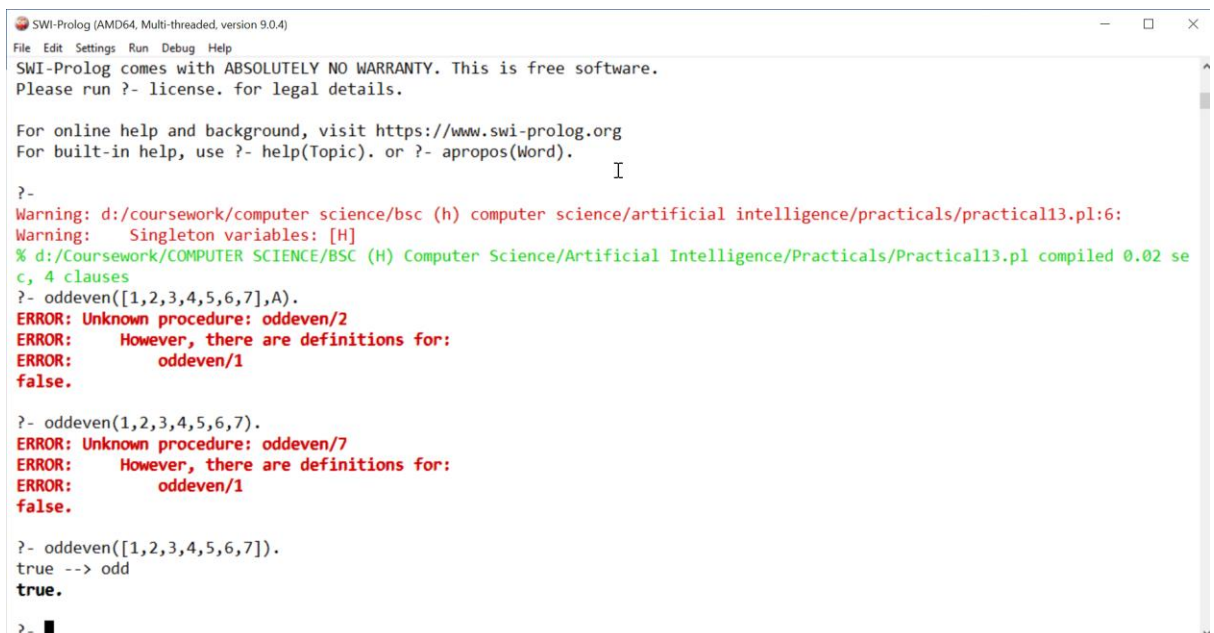


```
practical13.pl
File Edit Browse Compile Prolog Pce Help

practical13.pl
evenlength:-
    write('true --> even').
oddlength:-
    write('true --> odd').

oddeven([H|T]):-
    length(T,L),
    L>=0 ->
    (
        L1 is L+1,
        L2 is mod(L1,2),
        L2=:=0 ->
        evenlength
    ;
        oddlength
    ).
```

OUTPUT



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
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).

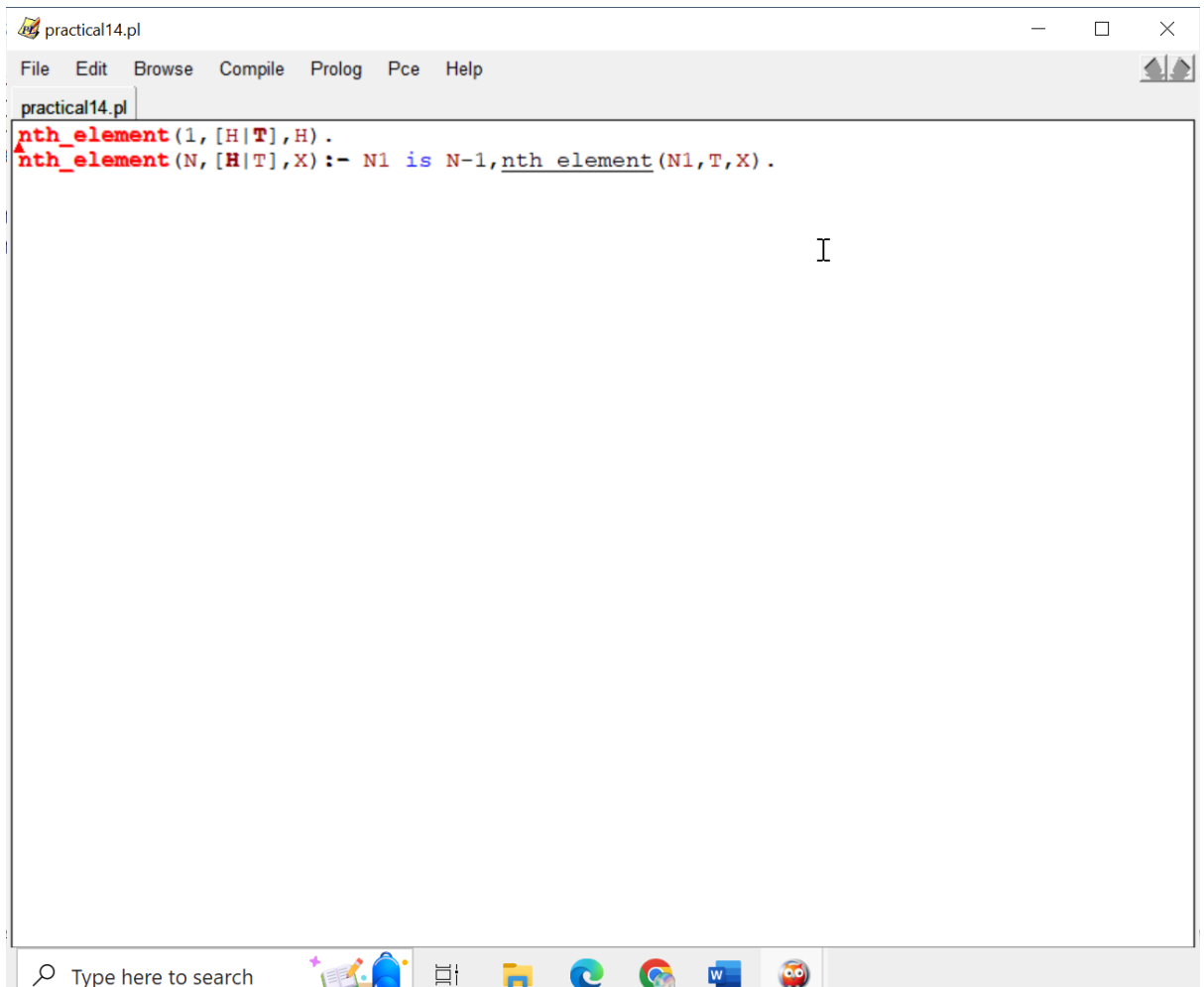
?-
Warning: d:/coursework/computer science/bsc (h) computer science/artificial intelligence/practicals/practical13.pl:6:
Warning: Singleton variables: [H]
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical13.pl compiled 0.02 se
c, 4 clauses
?- oddeven([1,2,3,4,5,6,7],A).
ERROR: Unknown procedure: oddeven/2
ERROR: However, there are definitions for:
ERROR: oddeven/1
false.

?- oddeven(1,2,3,4,5,6,7).
ERROR: Unknown procedure: oddeven/7
ERROR: However, there are definitions for:
ERROR: oddeven/1
false.

?- oddeven([1,2,3,4,5,6,7]).
true --> odd
true.

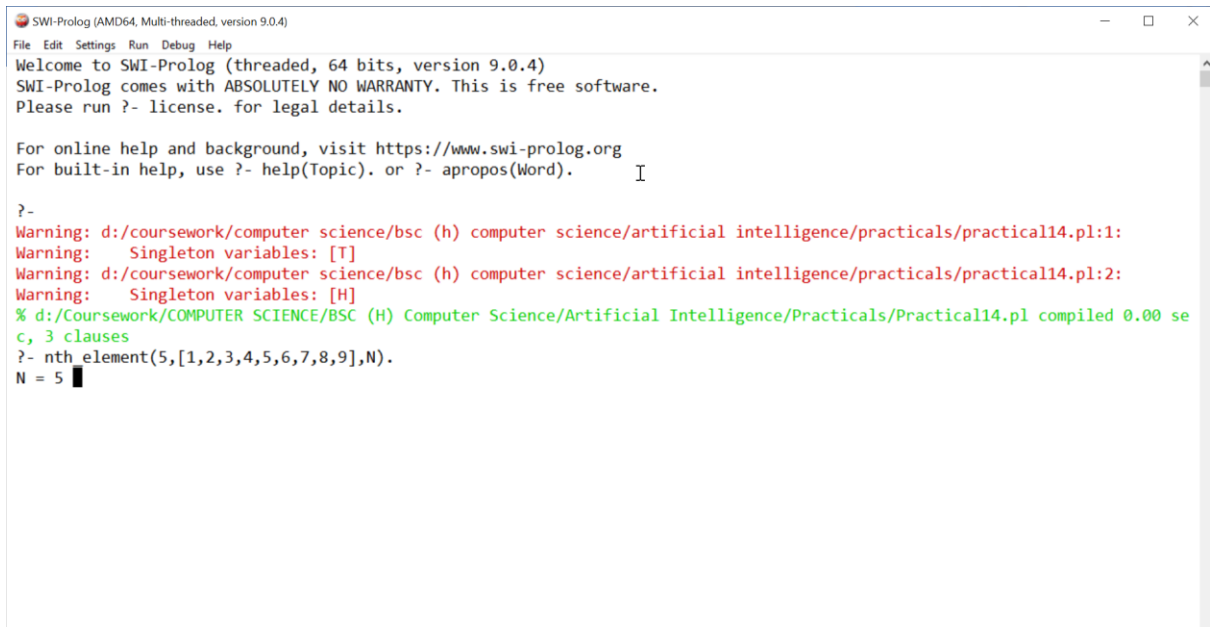
?-
```

14. 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 `N`th element of `L`.



```
practical14.pl
File Edit Browse Compile Prolog Pce Help
practical14.pl
nth_element(1, [H|T], H) .
nth_element(N, [H|T], X) :- N1 is N-1, nth_element(N1, T, X) .
```

OUTPUT

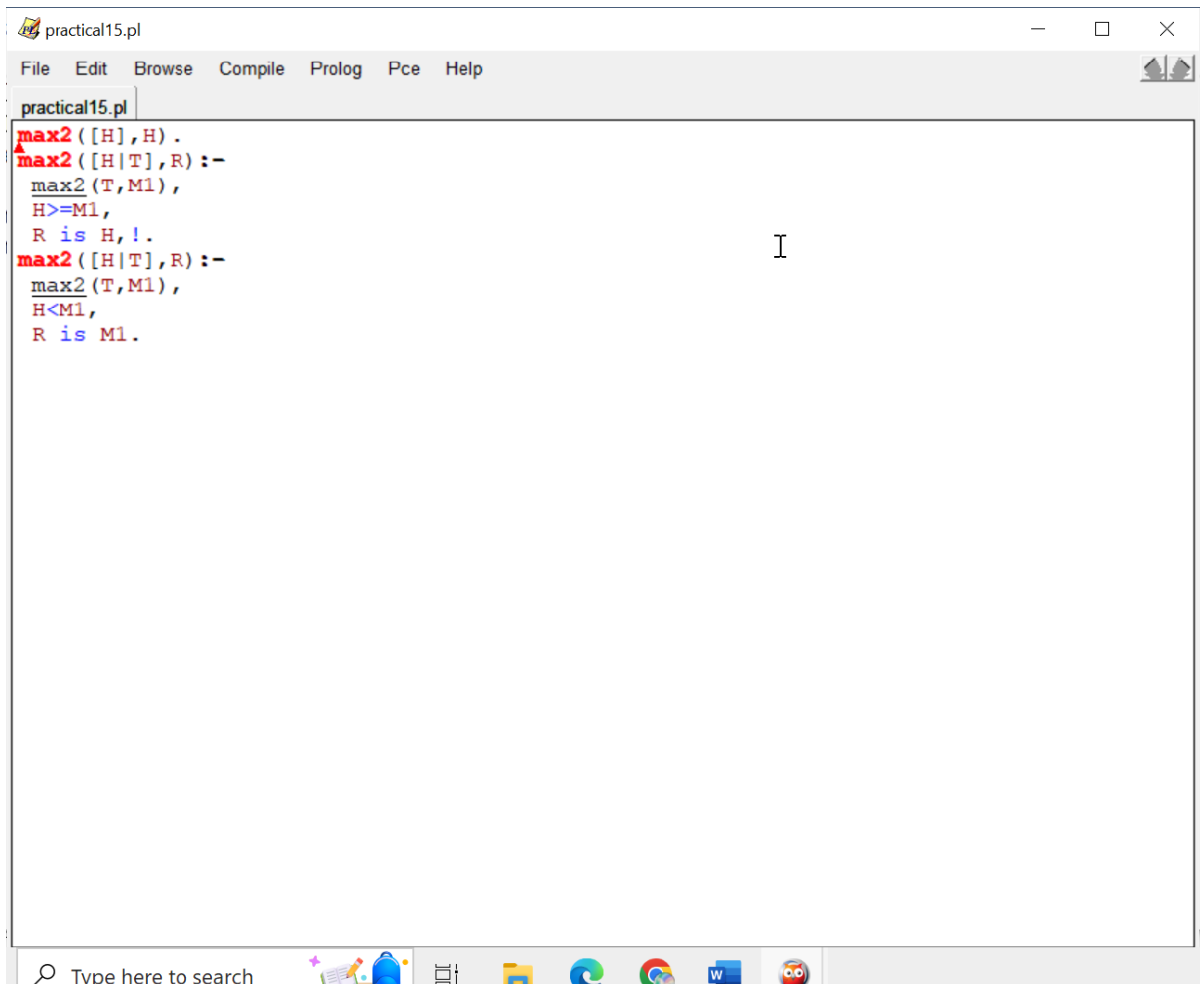


```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

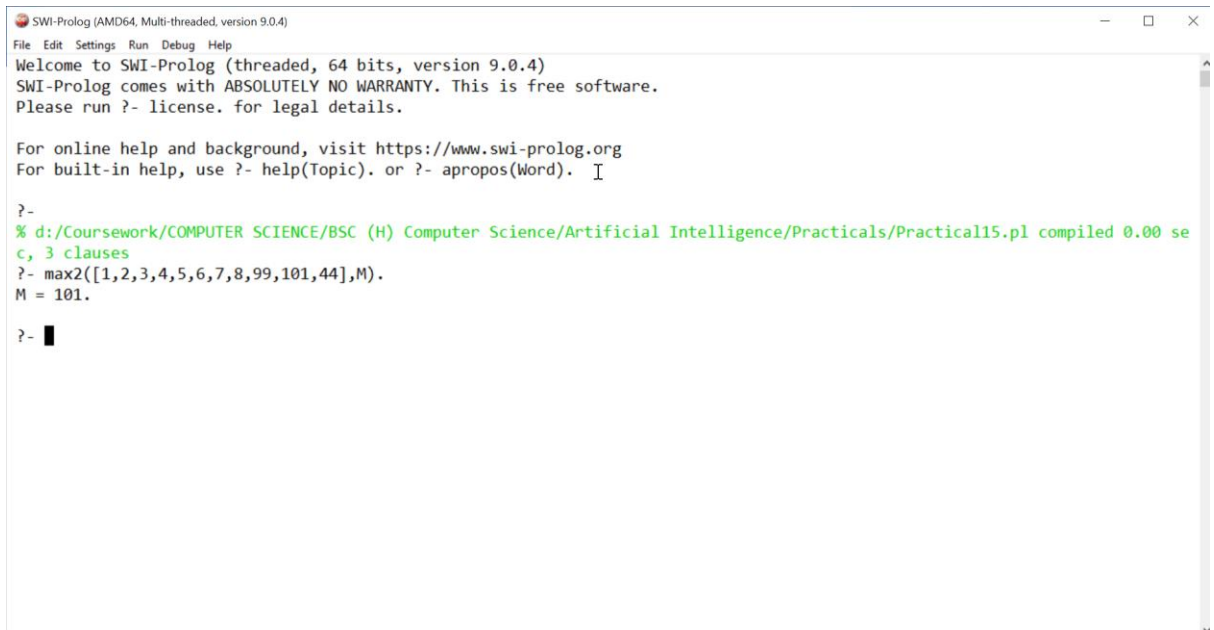
?-
Warning: d:/coursework/computer science/bsc (h) computer science/artificial intelligence/practicals/practical14.pl:1:
Warning: Singleton variables: [T]
Warning: d:/coursework/computer science/bsc (h) computer science/artificial intelligence/practicals/practical14.pl:2:
Warning: Singleton variables: [H]
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical14.pl compiled 0.00 se
c, 3 clauses
?- nth_element(5, [1,2,3,4,5,6,7,8,9], N).
N = 5
```

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



```
practical15.pl
File Edit Browse Compile Prolog Pce Help
practical15.pl
max2([H],H).
max2([H|T],R):-
    max2(T,M1),
    H>=M1,
    R is H,!
max2([H|T],R):-
    max2(T,M1),
    H<M1,
    R is M1.
```

OUTPUT



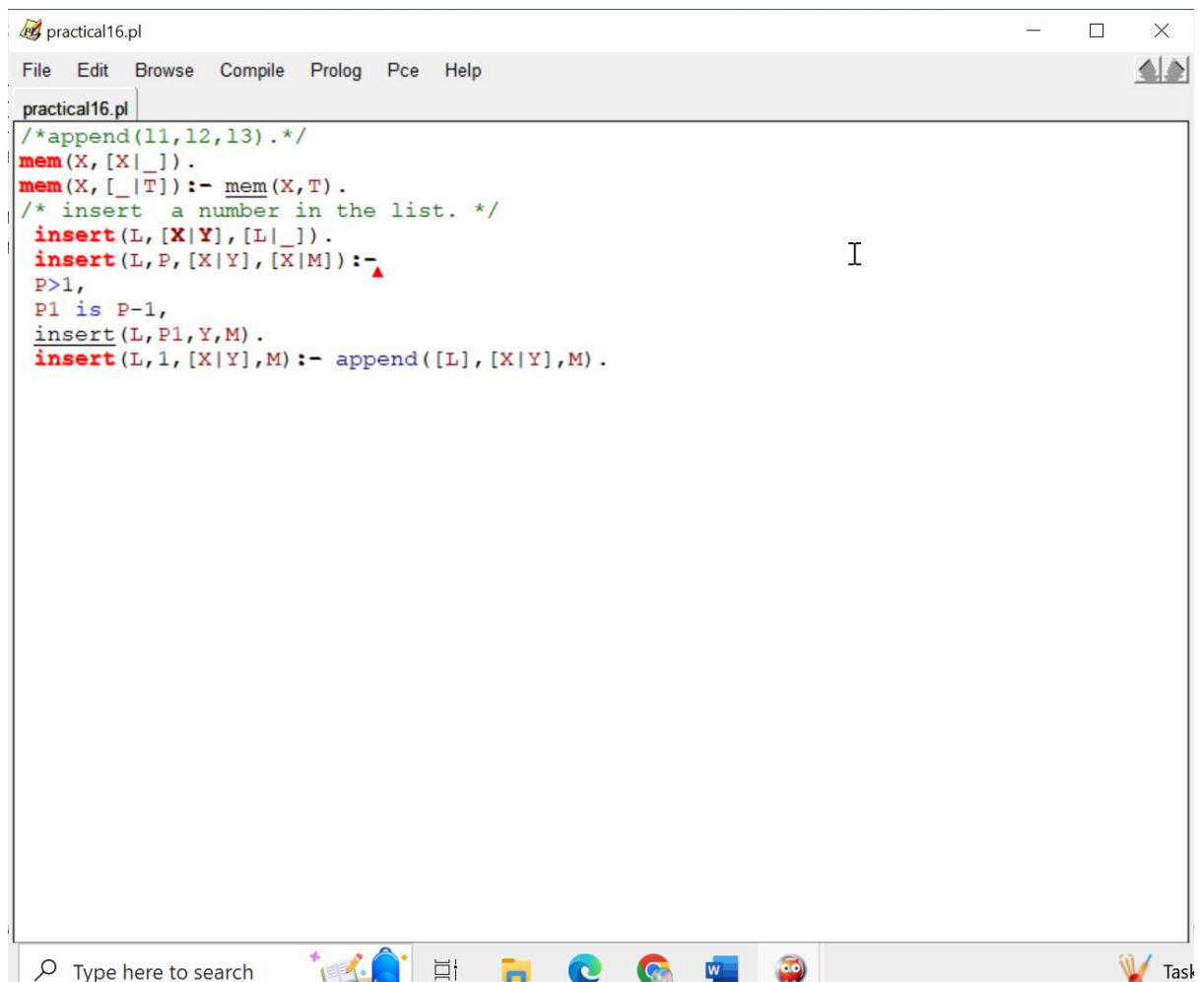
```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word). I

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical15.pl compiled 0.00 se
c, 3 clauses
?- max2([1,2,3,4,5,6,7,8,99,101,44],M).
M = 101.

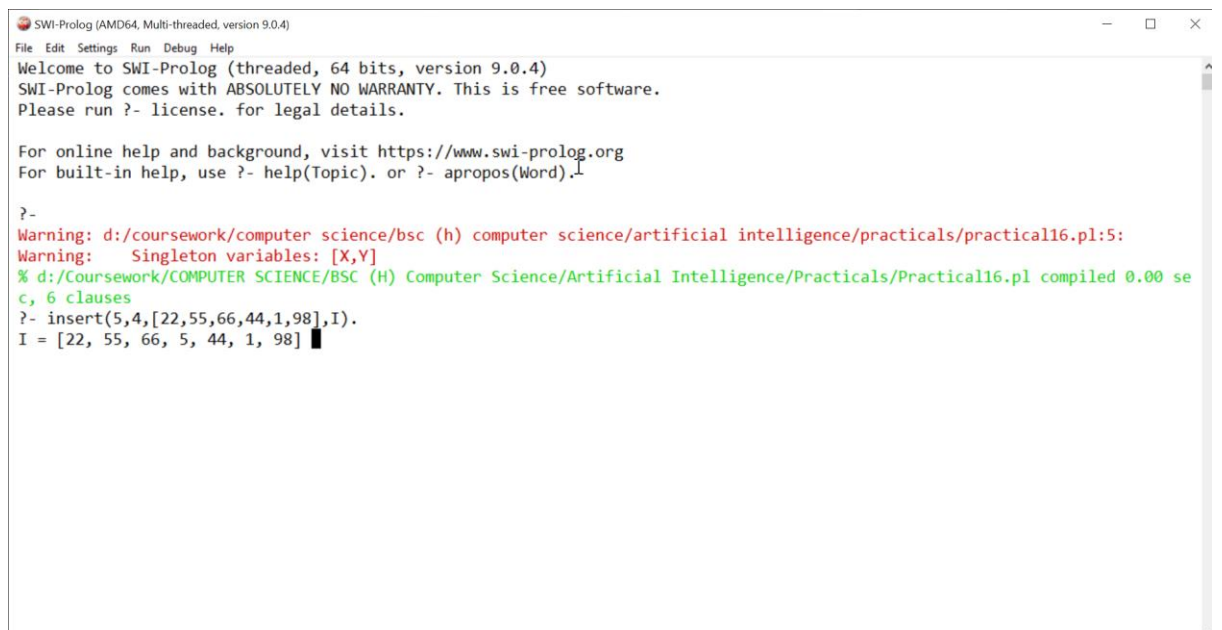
?-
```


16. 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



```
practical16.pl
File Edit Browse Compile Prolog Pce Help
practical16.pl
/*append(l1,l2,l3).*/
mem(X,[X|_]).
mem(X,[_|T]) :- mem(X,T).
/* insert a number in the list. */
insert(L,[X|Y],[L|_]).
insert(L,P,[X|Y],[X|M]) :-
P>1,
P1 is P-1,
insert(L,P1,Y,M).
insert(L,1,[X|Y],M) :- append([L],[X|Y],M).
```

OUTPUT

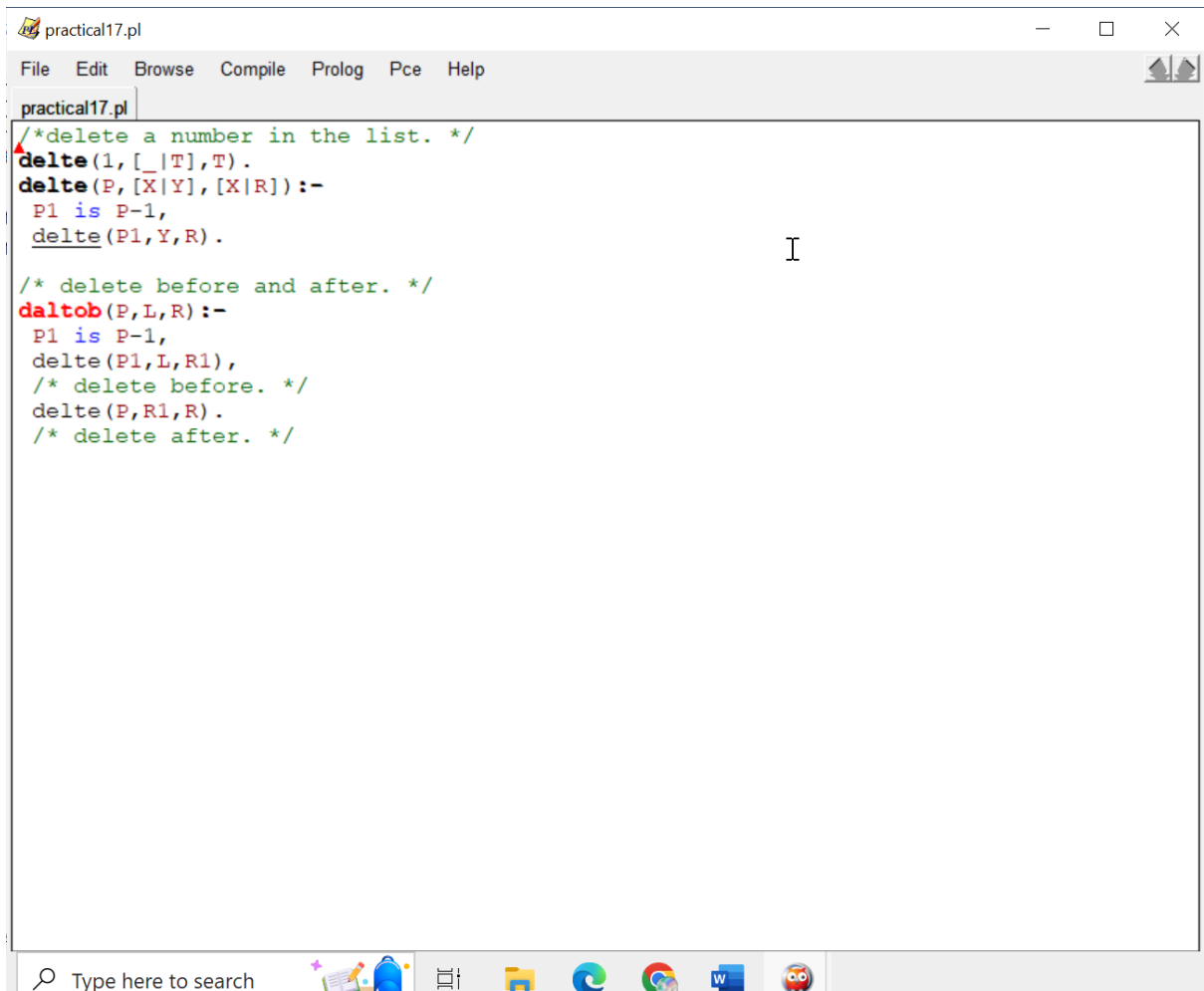


```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
Warning: d:/coursework/computer science/bsc (h) computer science/artificial intelligence/practicals/practical16.pl:5:
Warning: Singleton variables: [X,Y]
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical16.pl compiled 0.00 se
c, 6 clauses
?- insert(5,4,[22,55,66,44,1,98],I).
I = [22, 55, 66, 5, 44, 1, 98]
```

17. 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.

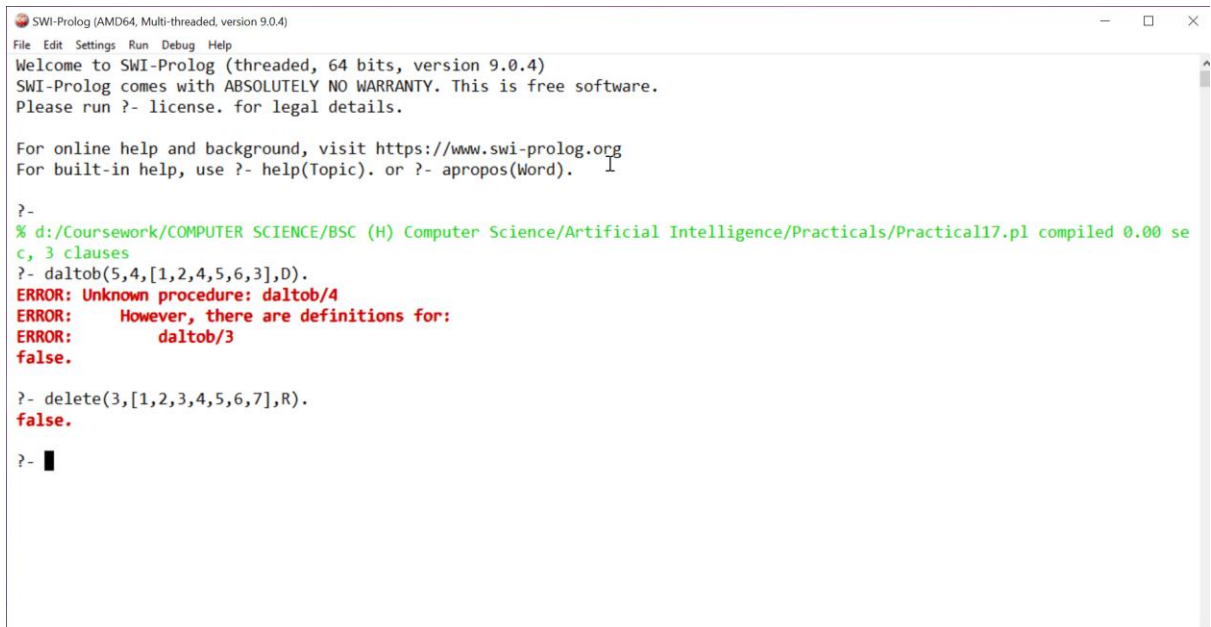


```
practical17.pl
File Edit Browse Compile Prolog Pce Help

/*delete a number in the list. */
delte(1, [_|T],T).
delte(P, [X|Y], [X|R]) :-
    P1 is P-1,
    delte(P1,Y,R).

/* delete before and after. */
daltoob(P,L,R) :-
    P1 is P-1,
    delte(P1,L,R1),
    /* delete before. */
    delte(P,R1,R).
    /* delete after. */
```

OUTPUT



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

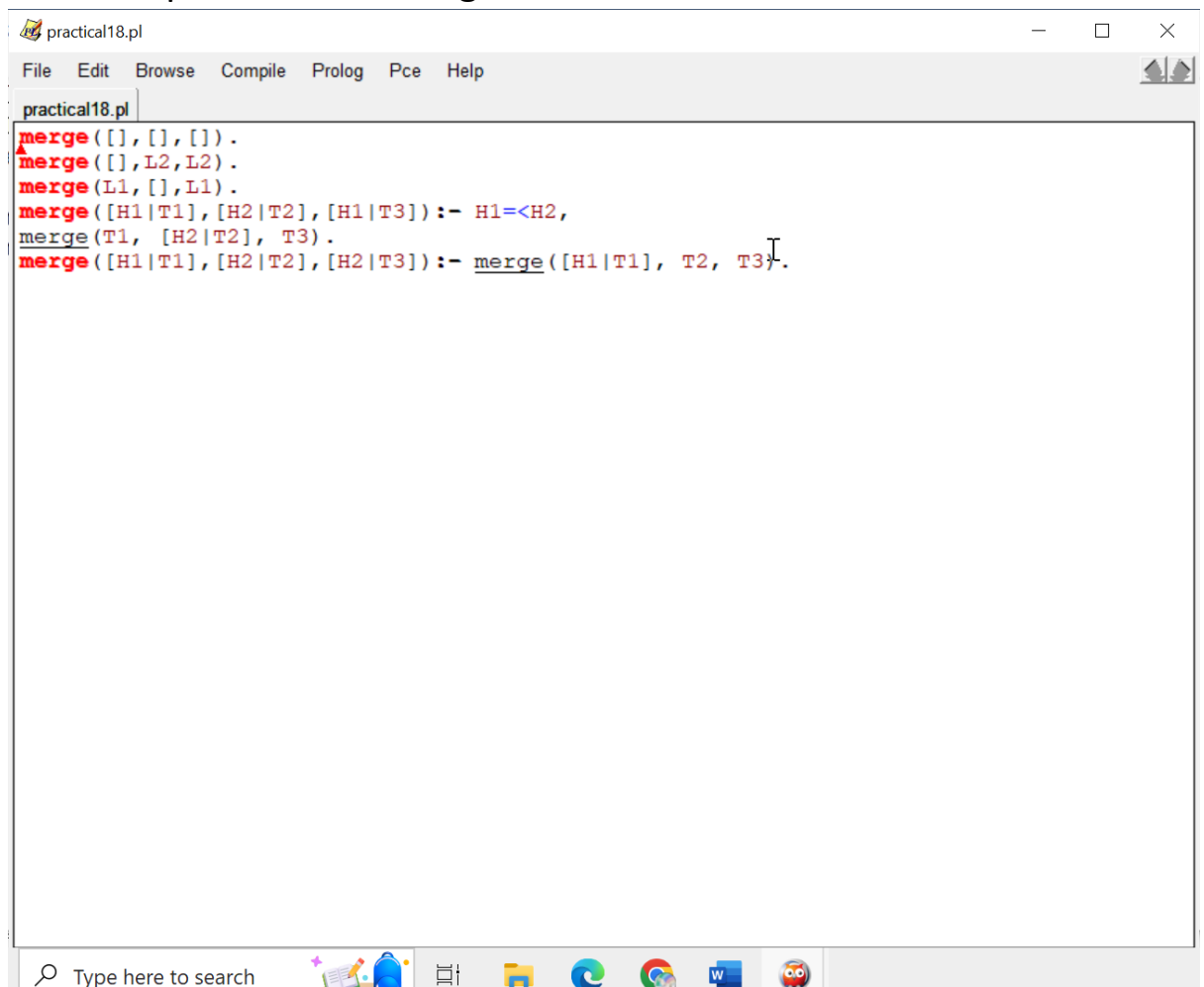
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical17.pl compiled 0.00 sec, 3 clauses
?- daltoob(5,4,[1,2,4,5,6,3],D).
ERROR: Unknown procedure: daltoob/4
ERROR: However, there are definitions for:
ERROR: daltoob/3
false.

?- delete(3,[1,2,3,4,5,6,7],R).
false.

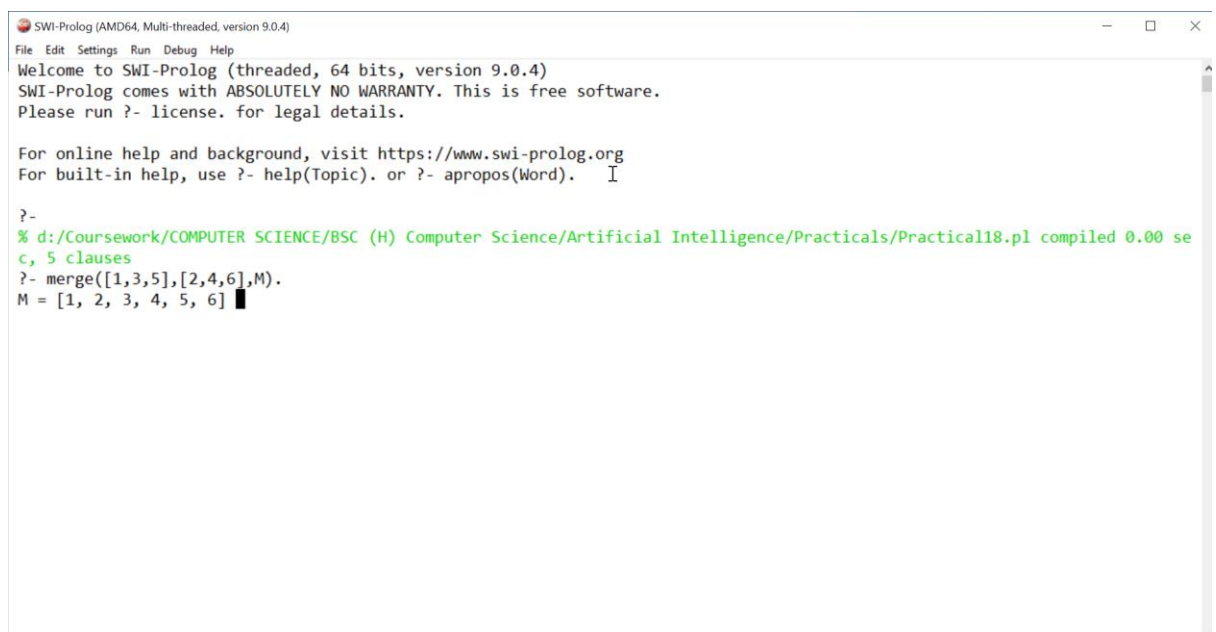
?-
```

18. Write a program in PROLOG to implement t merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list



```
practical18.pl
File Edit Browse Compile Prolog Pce Help
practical18.pl
merge([], [], []).
merge([], L2, L2).
merge(L1, [], L1).
merge([H1|T1], [H2|T2], [H1|T3]) :- H1 <= H2,
merge(T1, [H2|T2], T3).
merge([H1|T1], [H2|T2], [H2|T3]) :- merge([H1|T1], T2, T3).
```

OUTPUT



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
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).

?-
% d:/Coursework/COMPUTER SCIENCE/BSC (H) Computer Science/Artificial Intelligence/Practicals/Practical18.pl compiled 0.00 sec, 5 clauses
?- merge([1,3,5],[2,4,6],M).
M = [1, 2, 3, 4, 5, 6]
```