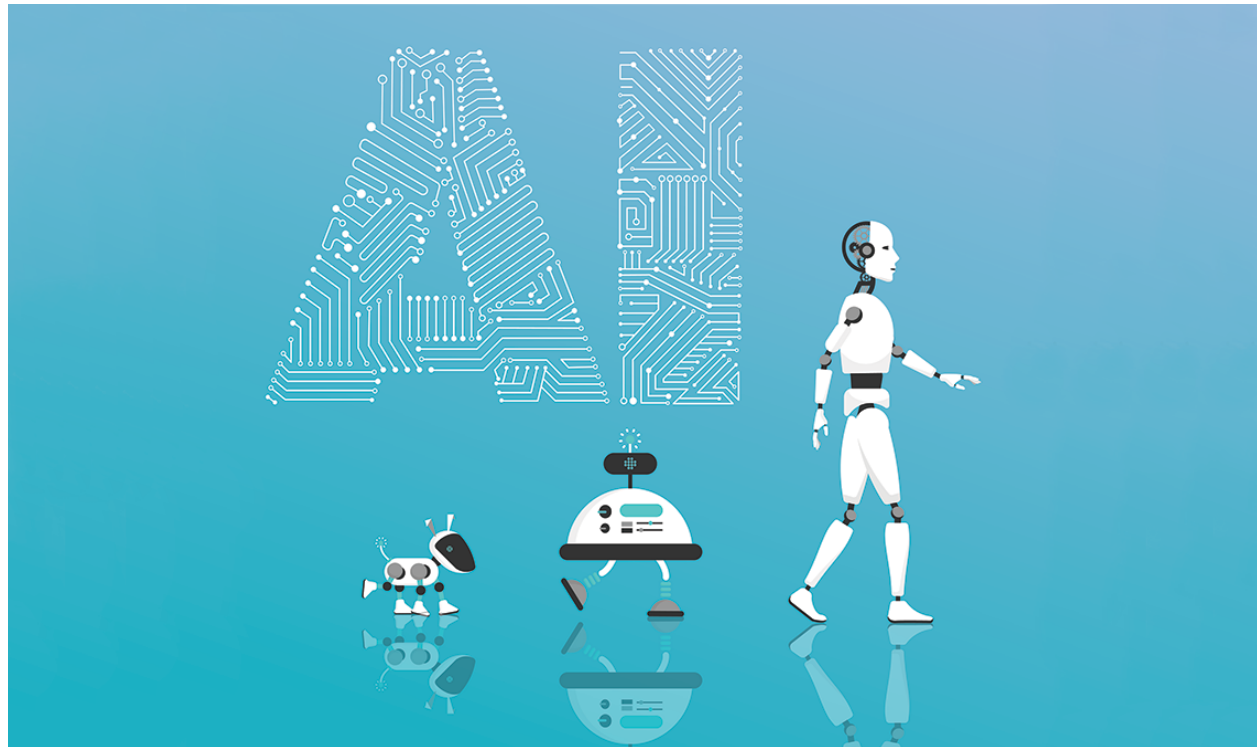




**Atma Ram Sanatan Dharma College**  
**University of Delhi**



# Artificial Intelligence

Submitted By

**KHUSHAL SACHDEVA**

20/88044 (College Roll)

20003570032 (University Roll)

BSc. (Hons.) Computer Science (SEM - 6)

Submitted To

**Ms Parul Jain Mam**

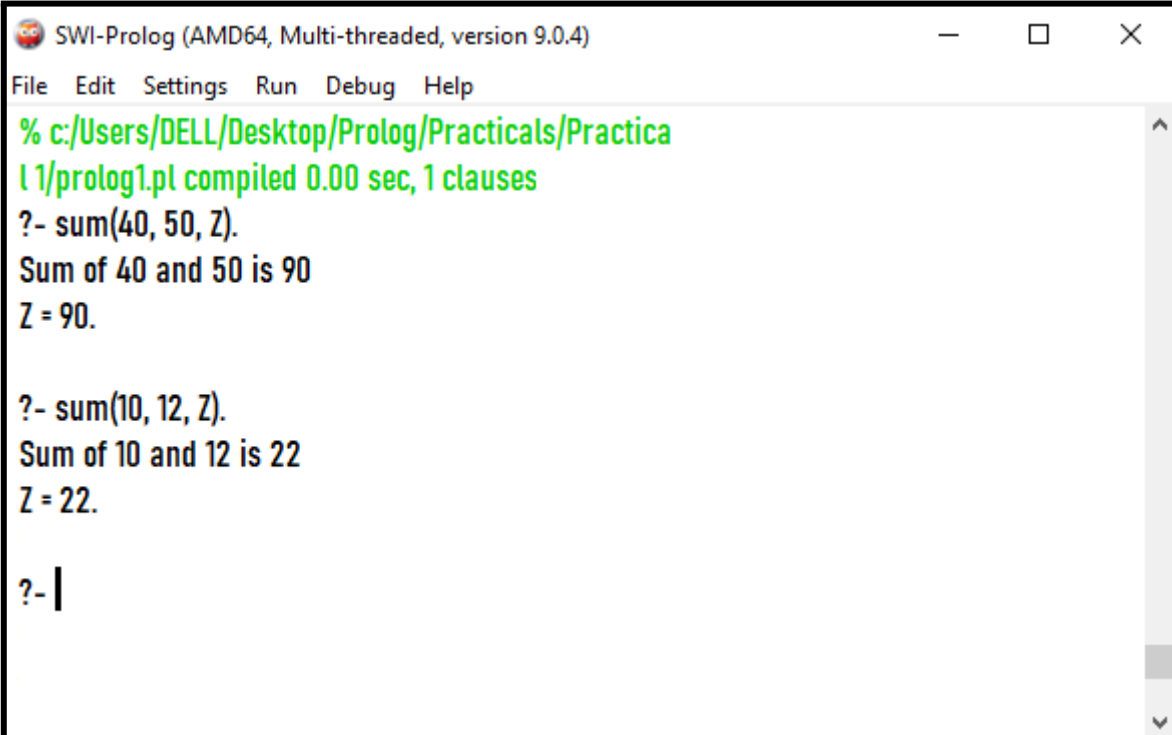
Department of Computer Science

# INDEX

Serial No.	Practical
1.	Write a prolog program to calculate the sum of two numbers.
2.	Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.
3.	Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.
4.	Write a program in PROLOG to implement generate_fib(N,T) where T represents the Nth term of the fibonacci series.
5.	Write a Prolog program to implement GCD of two numbers.
6.	Write a Prolog program to implement power (Num,Pow, Ans) : where Num is raised to the power Pow to get Ans.
7.	Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result.
8.	Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.
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.
10.	Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.
11.	Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.
12.	Write a Prolog program to implement sumlist(L, S) so that S is the sum of a given list L.
13.	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.
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 Nth element of L.
15.	Write a Prolog program to implement maxlist(L, M) so that M is the maximum number in the list.
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.
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.
18.	Write a program in PROLOG to implement merge (L1, L2, L3) where L1 is first ordered list and L2 is second ordered list and L3 represents the merged list.

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

```
sum(X, Y, Z):- Z is X + Y,  
    write('Sum of '),  
    write(X),  
    write(' and '),  
    write(Y),  
    write(' is '),  
    write(Z),!.
```

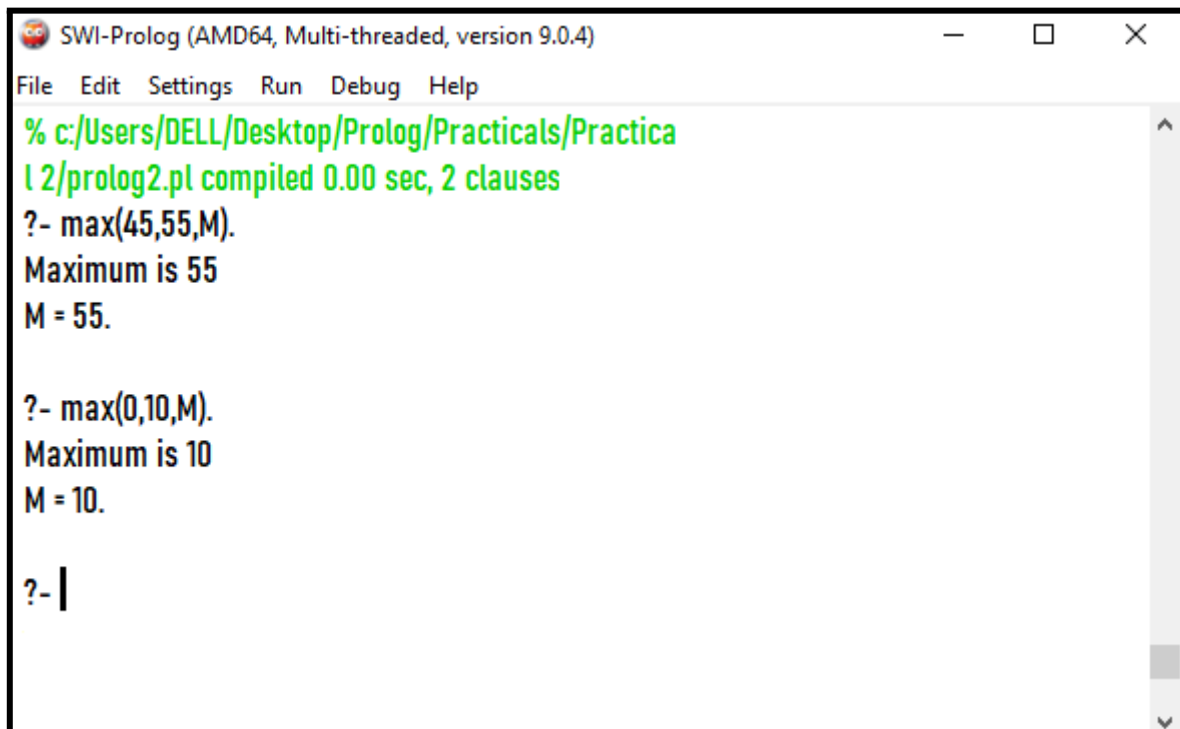


The screenshot shows a window titled "SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)". The menu bar includes "File", "Edit", "Settings", "Run", "Debug", and "Help". The main text area displays the following output:

```
% c:/Users/DELL/Desktop/Prolog/Practicals/Practica  
l 1/prolog1.pl compiled 0.00 sec, 1 clauses  
?- sum(40, 50, Z).  
Sum of 40 and 50 is 90  
Z = 90.  
  
?- sum(10, 12, Z).  
Sum of 10 and 12 is 22  
Z = 22.  
  
?- |
```

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

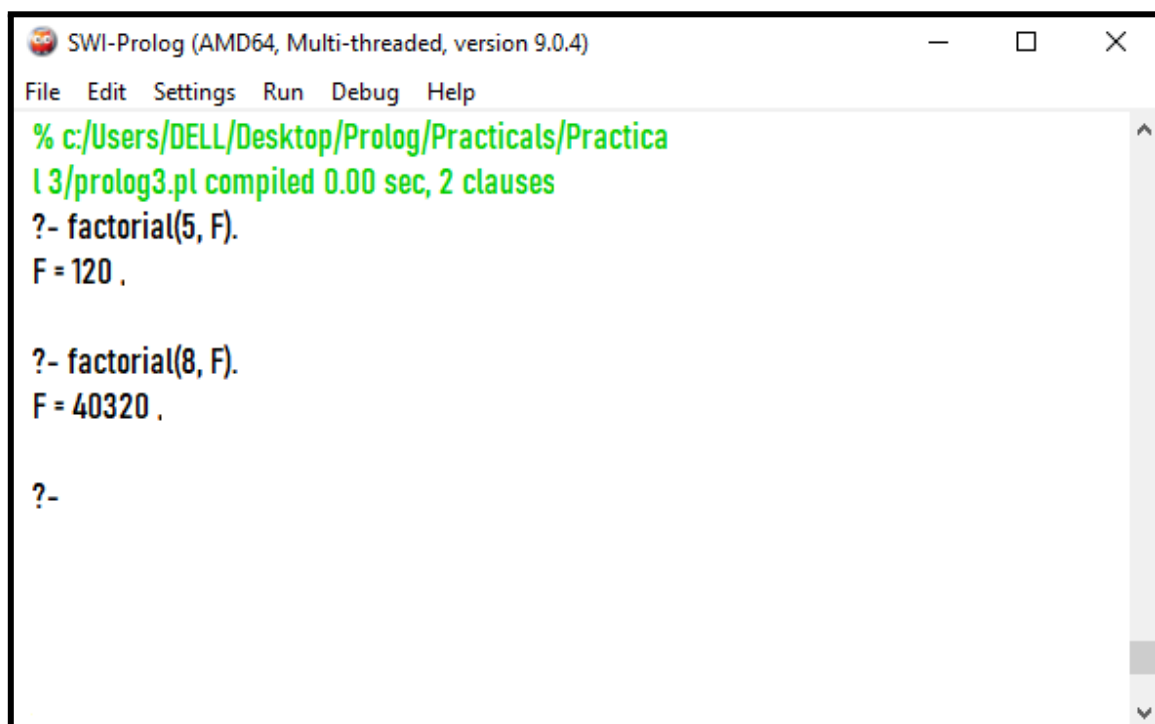
```
max(X, Y, M):-  
    X>Y, M is X,  
    write('Maximum is '),  
    write(M).  
  
max(X, Y, M):-  
    Y>=X, M is Y,  
    write('Maximum is '),  
    write(M).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)  
File Edit Settings Run Debug Help  
% c:/Users/DELL/Desktop/Prolog/Practicals/Practica  
l 2/prolog2.pl compiled 0.00 sec, 2 clauses  
?- max(45,55,M).  
Maximum is 55  
M = 55.  
  
?- max(0,10,M).  
Maximum is 10  
M = 10.  
  
?- |
```

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

```
factorial(0, 1).  
  
factorial(N, F):-  
    N1 is N-1,  
    factorial(N1, Y),  
    F is Y*N.
```

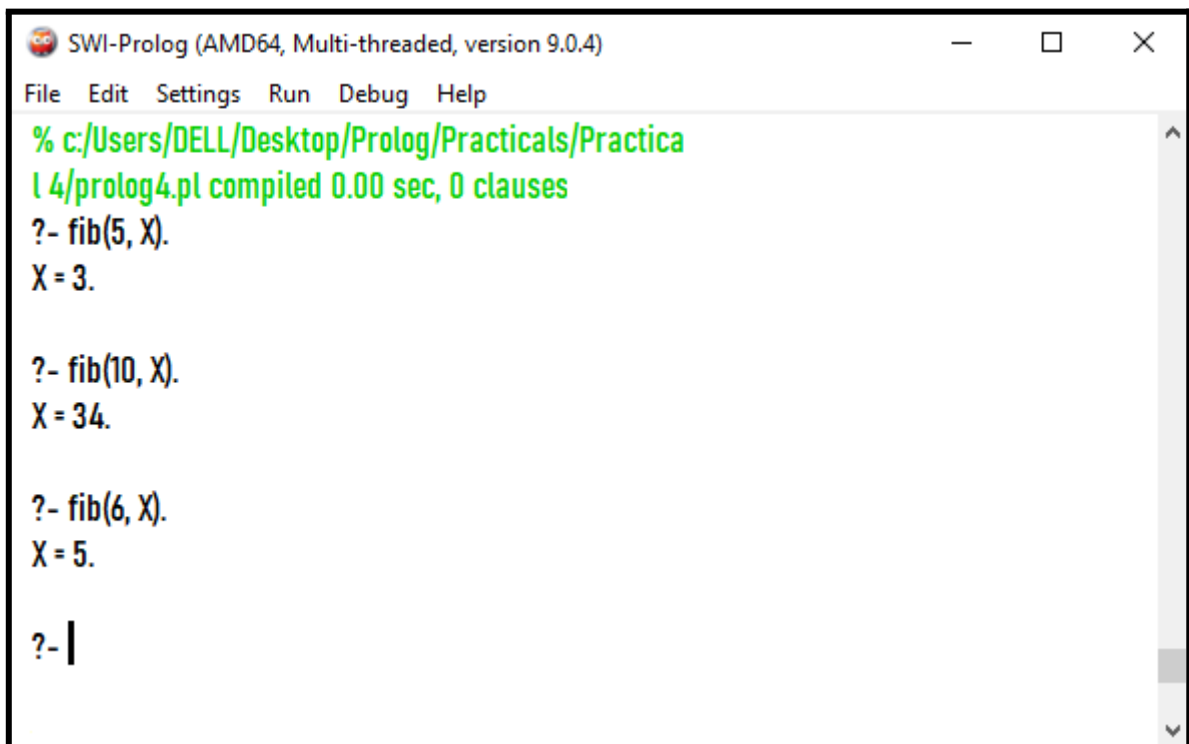


The screenshot shows a window titled "SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)". The menu bar includes "File", "Edit", "Settings", "Run", "Debug", and "Help". The main text area displays the following output in green:

```
% c:/Users/DELL/Desktop/Prolog/Practicals/Practica  
l 3/prolog3.pl compiled 0.00 sec, 2 clauses  
?- factorial(5, F).  
F = 120 .  
  
?- factorial(8, F).  
F = 40320 .  
  
?-
```

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

```
fib(1,0).  
fib(2,1).  
  
fib(N,X):-  
    N1 is N-1,  
    fib(N1,X1),  
    N2 is N-2,  
    fib(N2,X2),  
    X is X1+X2,!.
```

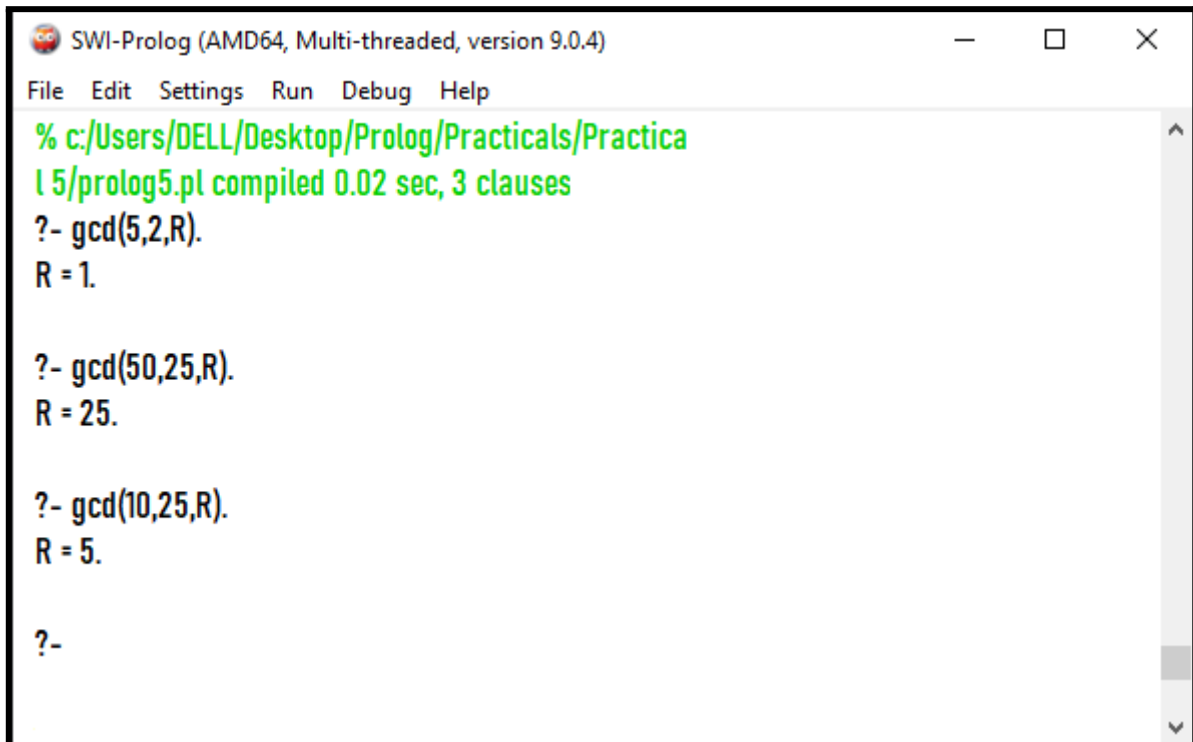


The screenshot shows a window titled "SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)". The menu bar includes "File", "Edit", "Settings", "Run", "Debug", and "Help". The main text area displays the following output:

```
% c:/Users/DELL/Desktop/Prolog/Practicals/Practica  
l 4/prolog4.pl compiled 0.00 sec, 0 clauses  
?- fib(5, X).  
X = 3.  
  
?- fib(10, X).  
X = 34.  
  
?- fib(6, X).  
X = 5.  
  
?- |
```

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

```
gcd(0,A,A):-!.  
gcd(A,0,A):-!.  
gcd(A,B,R):-B1 is mod(A,B),gcd(B,B1,R).
```

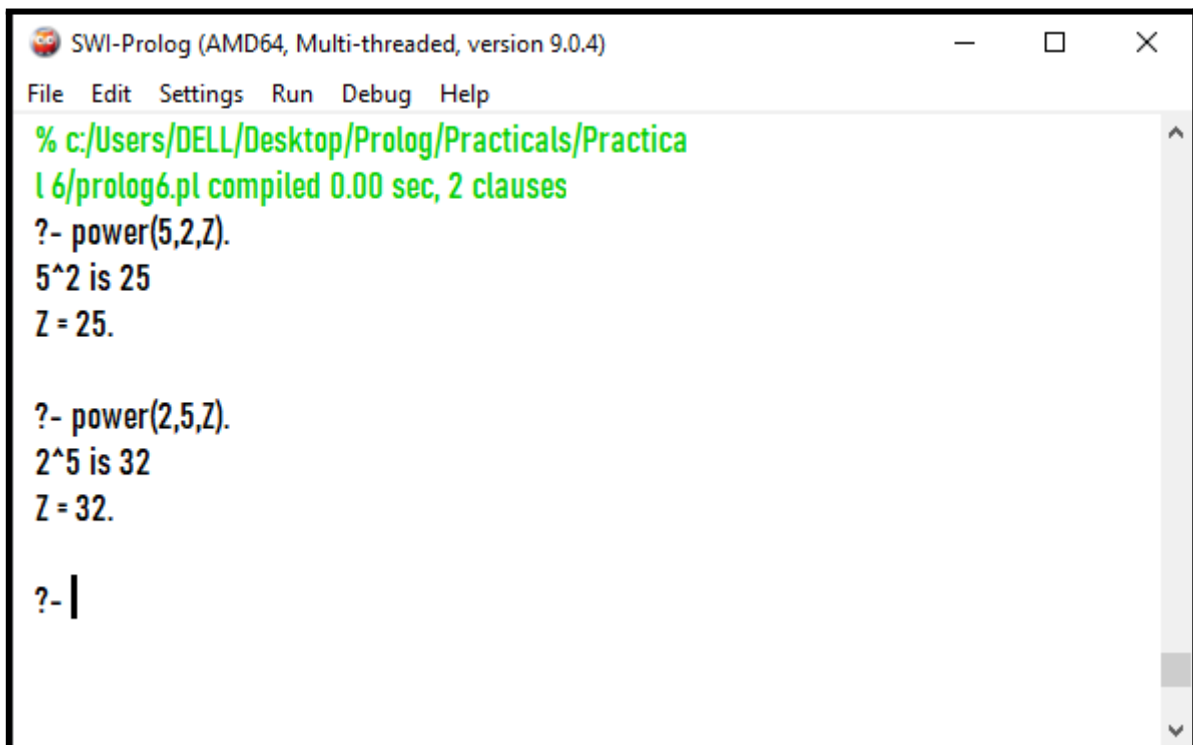


The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following content:

```
% c:/Users/DELL/Desktop/Prolog/Practicals/Practica  
l 5/prolog5.pl compiled 0.02 sec, 3 clauses  
?- gcd(5,2,R).  
R = 1.  
  
?- gcd(50,25,R).  
R = 25.  
  
?- gcd(10,25,R).  
R = 5.  
  
?-
```

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

```
power(X,0):- !.  
power(Num,Pow,Ans):-  
    Ans is Num^Pow,  
    write(Num),  
    write('^'),  
    write(Pow),  
    write(' is '),  
    write(Ans),!.
```



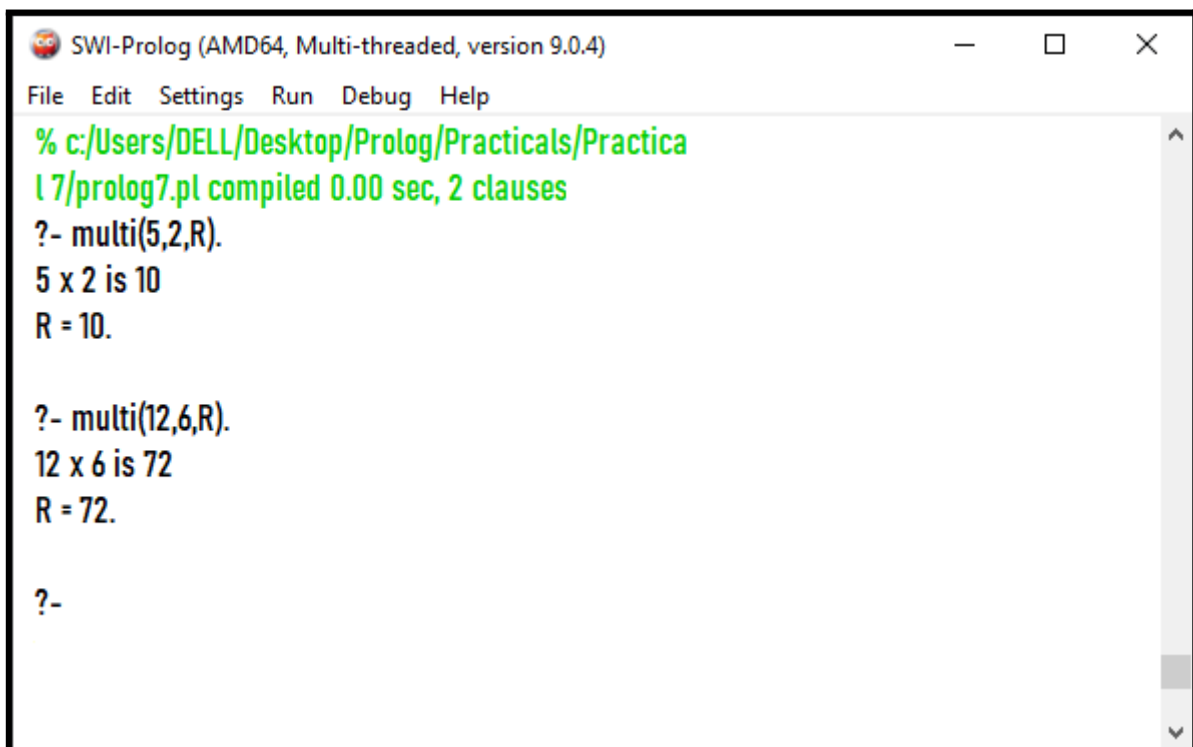
The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following output:

```
% c:/Users/DELL/Desktop/Prolog/Practicals/Practica  
l 6/prolog6.pl compiled 0.00 sec, 2 clauses  
?- power(5,2,Z).  
5^2 is 25  
Z = 25.  
  
?- power(2,5,Z).  
2^5 is 32  
Z = 32.  
  
?- |
```



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

```
multi(X,0).  
multi(N1,N2,R):-  
    R is N1*N2,  
    write(N1),  
    write(' x '),  
    write(N2),  
    write(' is '),  
    write(R),!.
```

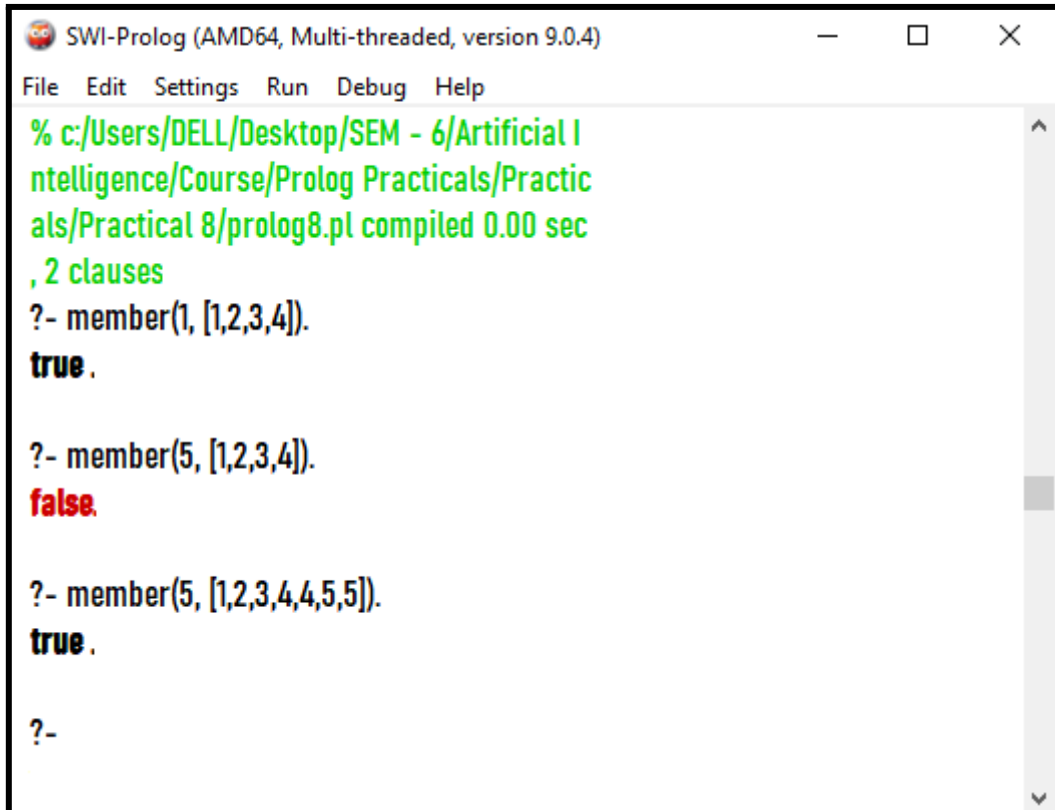


The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following output:

```
% c:/Users/DELL/Desktop/Prolog/Practicals/Practica  
l 7/prolog7.pl compiled 0.00 sec, 2 clauses  
?- multi(5,2,R).  
5 x 2 is 10  
R = 10.  
  
?- multi(12,6,R).  
12 x 6 is 72  
R = 72.  
  
?-
```

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

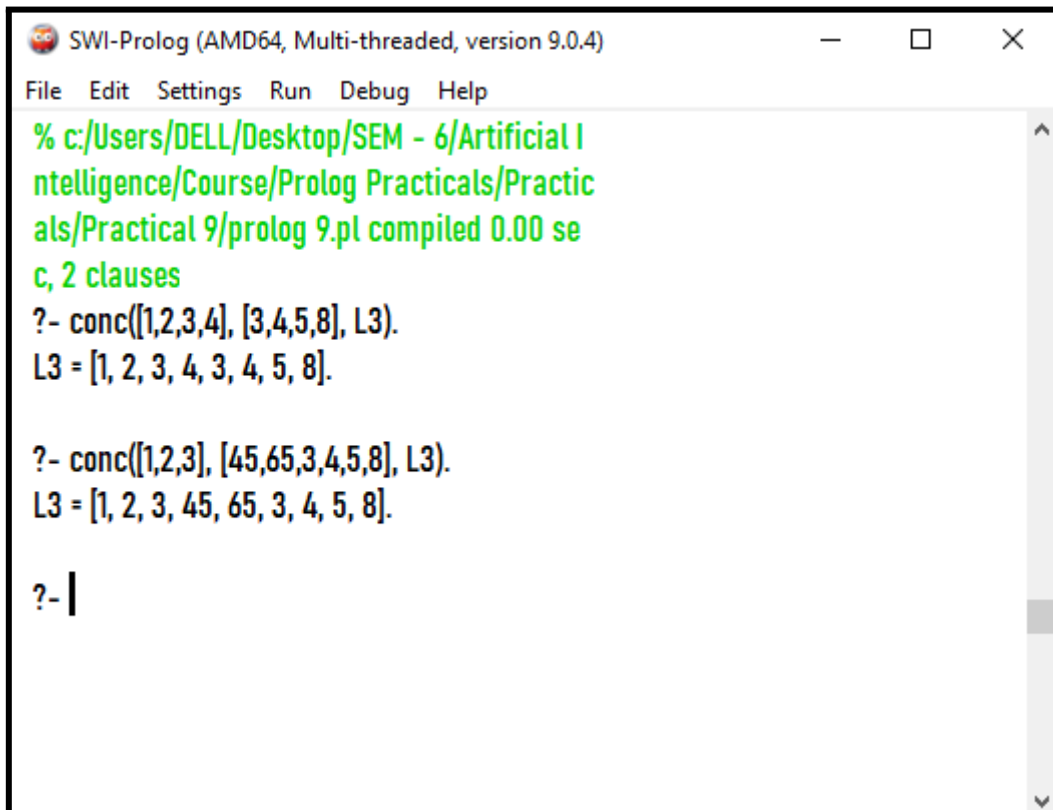
```
member(X, [X|_]).  
member(X, [_|L]):- member(X, L).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)  
File Edit Settings Run Debug Help  
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 8/prolog8.pl compiled 0.00 sec  
, 2 clauses  
?- member(1, [1,2,3,4]).  
true.  
  
?- member(5, [1,2,3,4]).  
false.  
  
?- member(5, [1,2,3,4,4,5,5]).  
true.  
  
?-
```

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`.

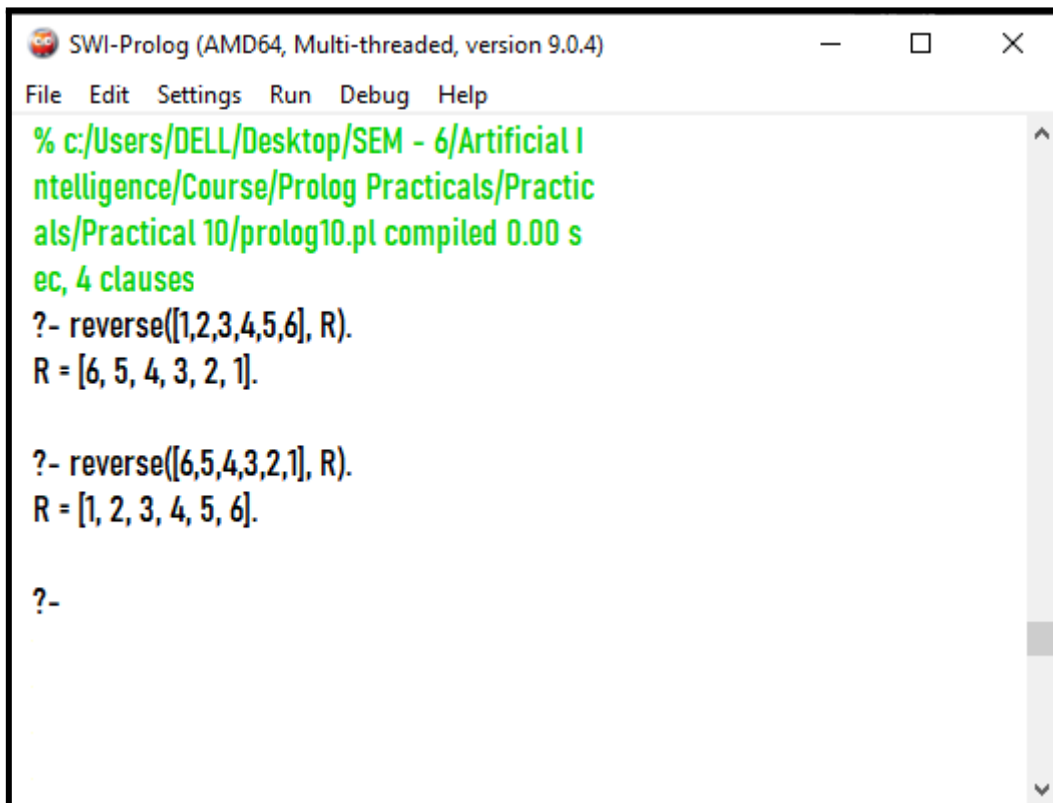
```
conc([], List, List).  
conc([X|L1], L2, [X|L3]):- conc(L1, L2, L3).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)  
File Edit Settings Run Debug Help  
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 9/prolog 9.pl compiled 0.00 sec, 2 clauses  
?- conc([1,2,3,4], [3,4,5,8], L3).  
L3 = [1, 2, 3, 4, 3, 4, 5, 8].  
  
?- conc([1,2,3], [45,65,3,4,5,8], L3).  
L3 = [1, 2, 3, 45, 65, 3, 4, 5, 8].  
  
?- |
```

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

```
append([],L,L).
append([X|L1],L2,[X|L3]):- append(L1,L2,L3).
reverse([],[]).
reverse([H|T],R):-reverse(T,L1),append(L1,[H],R).
```



The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following output:

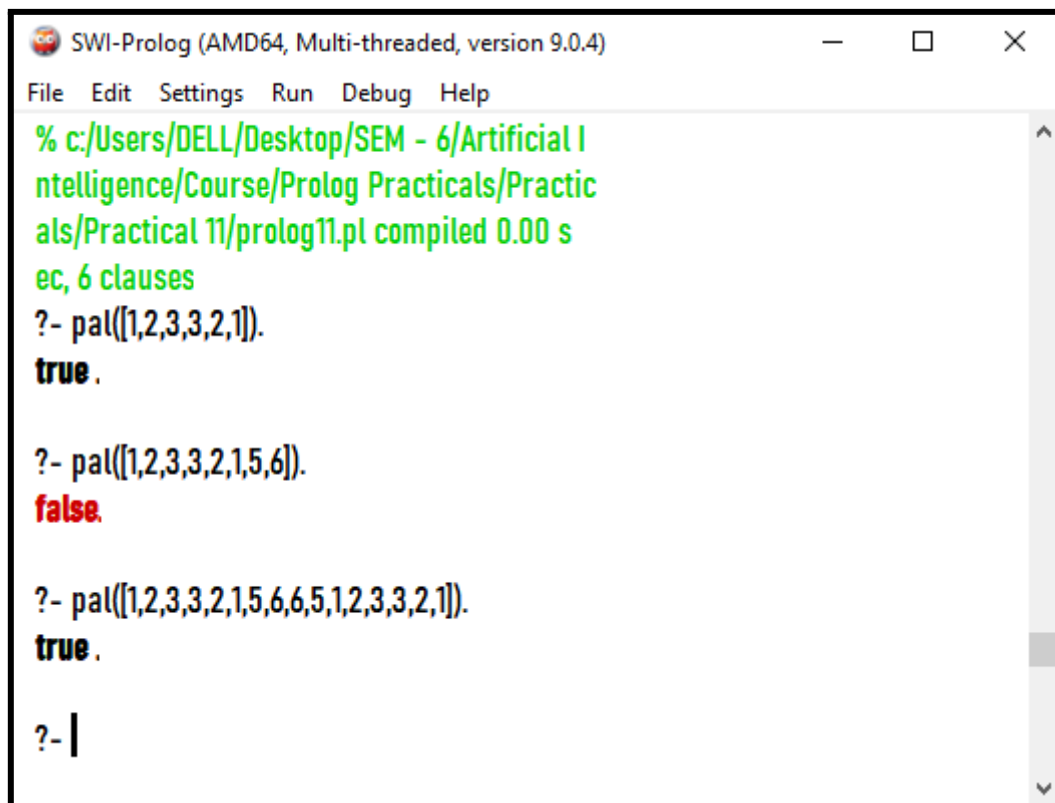
```
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 10/prolog10.pl compiled 0.00 sec, 4 clauses
?- reverse([1,2,3,4,5,6], R).
R = [6, 5, 4, 3, 2, 1].

?- reverse([6,5,4,3,2,1], R).
R = [1, 2, 3, 4, 5, 6].

?-
```

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

```
append([],L,L).
append([X|L1],L2,[X|L3]):- append(L1,L2,L3).
pal([]).
pal([_]).
pal(Plist):-append([H|T],[H],Plist),pal(T).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 11/prolog11.pl compiled 0.00 sec, 6 clauses
?- pal([1,2,3,3,2,1]).
true.

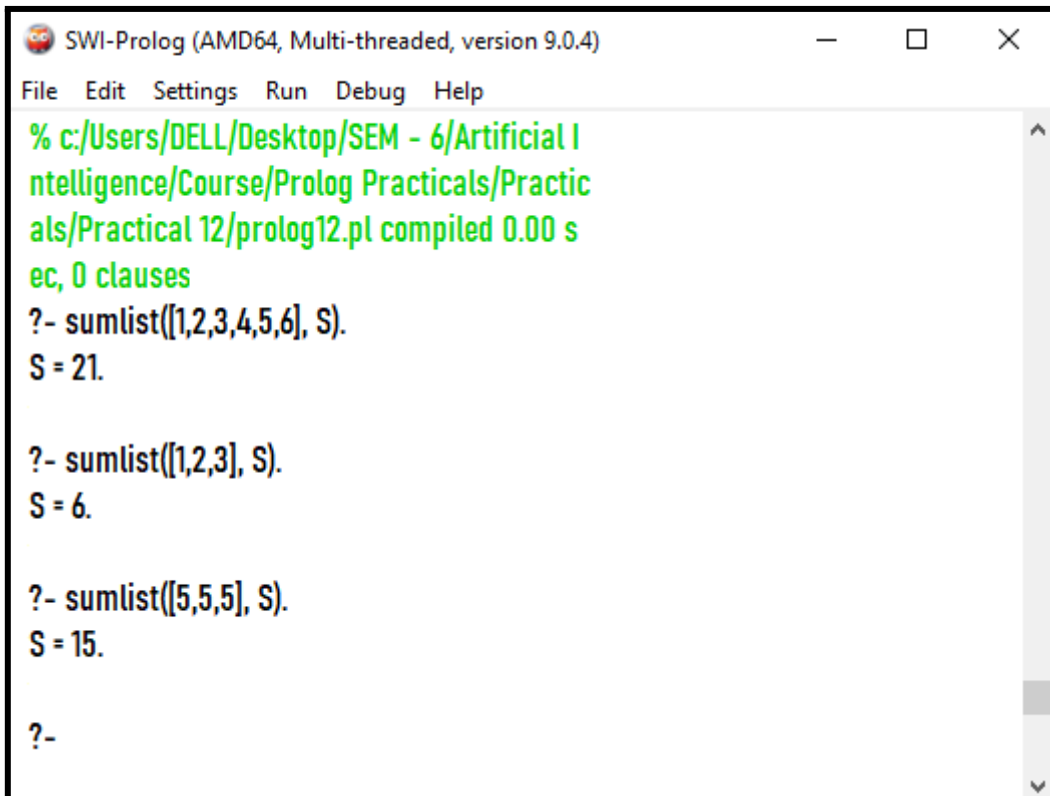
?- pal([1,2,3,3,2,1,5,6]).
false.

?- pal([1,2,3,3,2,1,5,6,6,5,1,2,3,3,2,1]).
true.

?- |
```

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

```
sumlist([],0).  
sumlist([H|T],S):- sumlist(T,S1), S is H+S1.
```

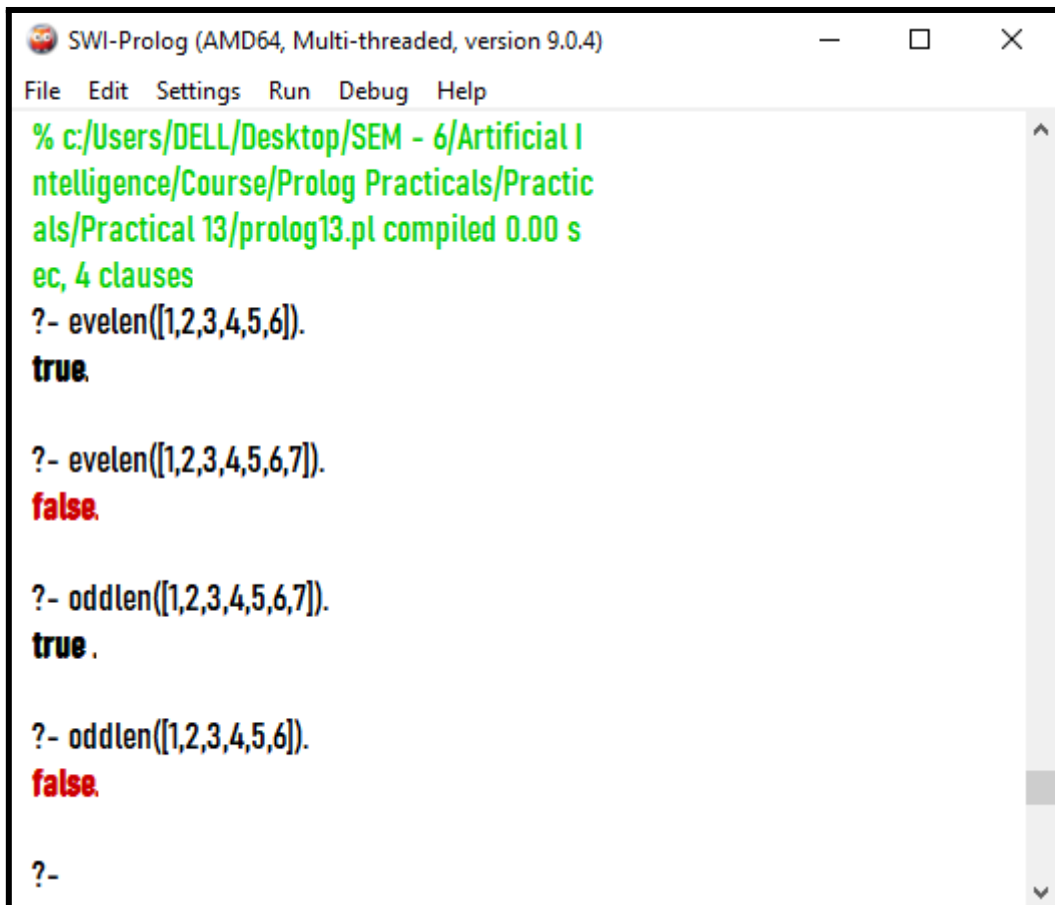


The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following output:

```
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 12/prolog12.pl compiled 0.00 sec, 0 clauses  
?- sumlist([1,2,3,4,5,6], S).  
S = 21.  
  
?- sumlist([1,2,3], S).  
S = 6.  
  
?- sumlist([5,5,5], S).  
S = 15.  
  
?-
```

**13. 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.**

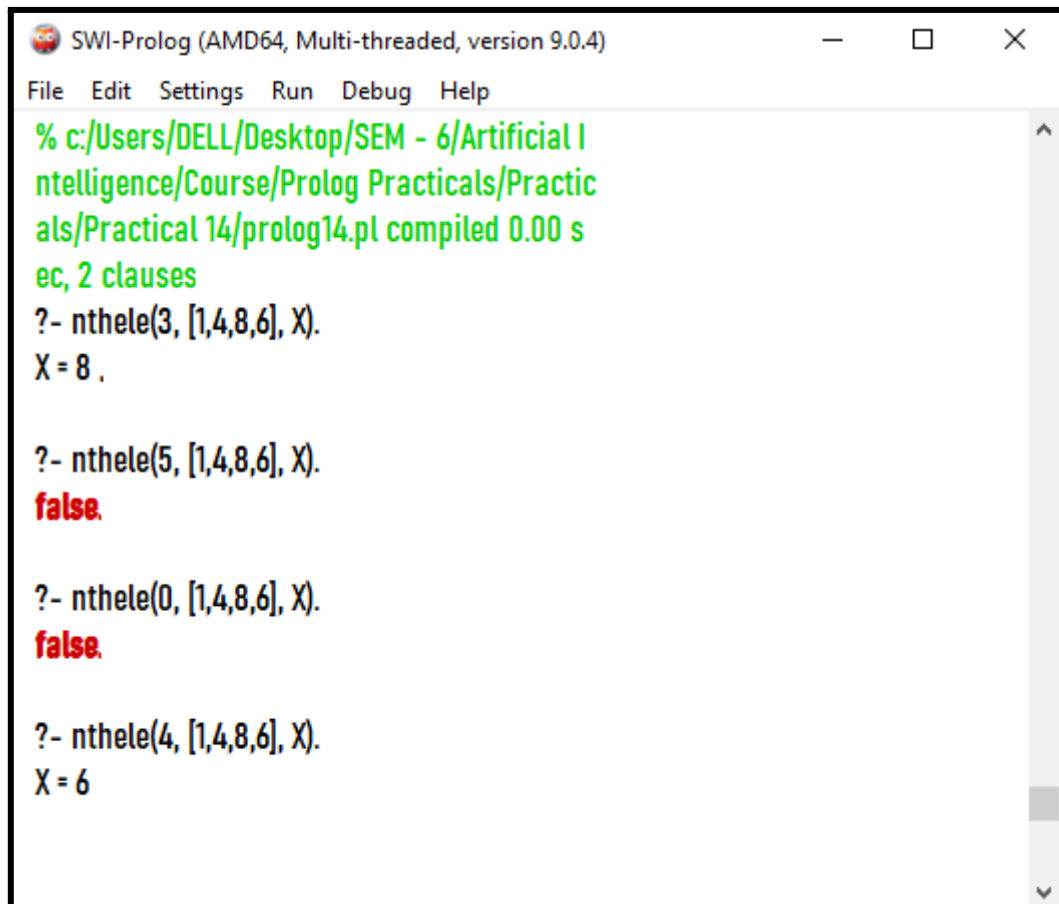
```
evelen([]).  
evelen([_|[_|List]]):- evelen(List).  
oddlen([_]).  
oddlen([_|[_|List]]):- oddlen(List).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)  
File Edit Settings Run Debug Help  
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 13/prolog13.pl compiled 0.00 sec, 4 clauses  
?- evelen([1,2,3,4,5,6]).  
true.  
  
?- evelen([1,2,3,4,5,6,7]).  
false.  
  
?- oddlen([1,2,3,4,5,6,7]).  
true.  
  
?- oddlen([1,2,3,4,5,6]).  
false.  
  
?-
```

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 Nth element of L.

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



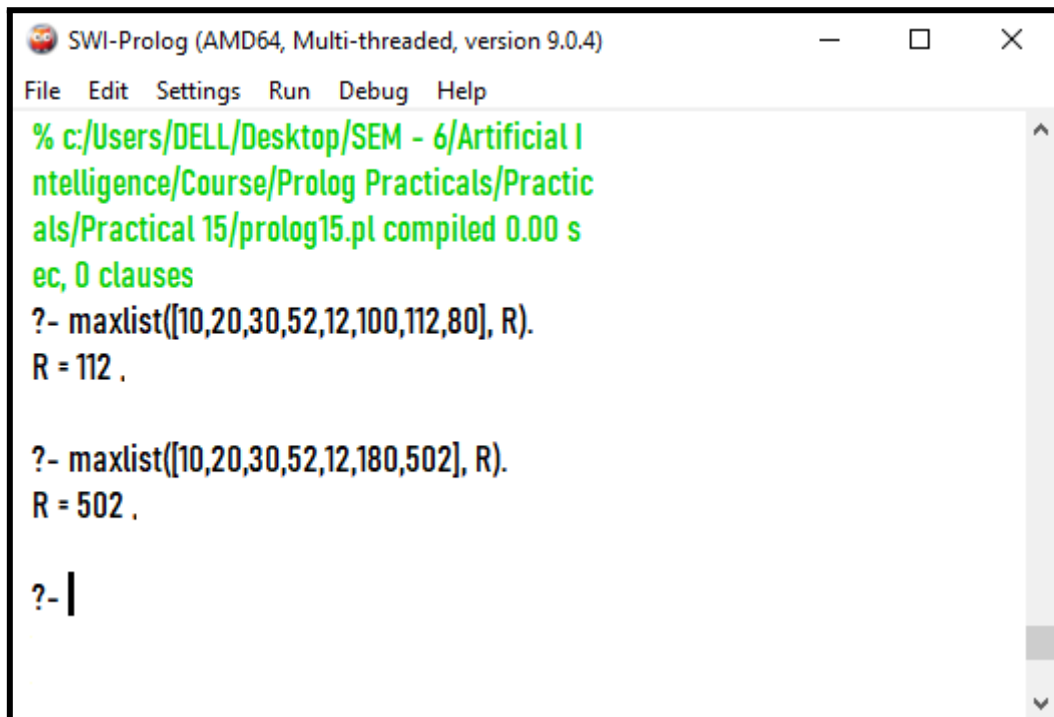
The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following content:

```
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 14/prolog14.pl compiled 0.00 s  
ec, 2 clauses  
?- nthele(3, [1,4,8,6], X).  
X = 8 .  
  
?- nthele(5, [1,4,8,6], X).  
false.  
  
?- nthele(0, [1,4,8,6], X).  
false.  
  
?- nthele(4, [1,4,8,6], X).  
X = 6
```



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

```
max(X,Y,Z):- X>Y, Z is X.  
max(X,Y,Z):- X<=Y, Z is Y.  
maxlist([],0).  
maxlist([R],R).  
maxlist([H|T],R):- maxlist(T,R1), max(H,R1,R).
```

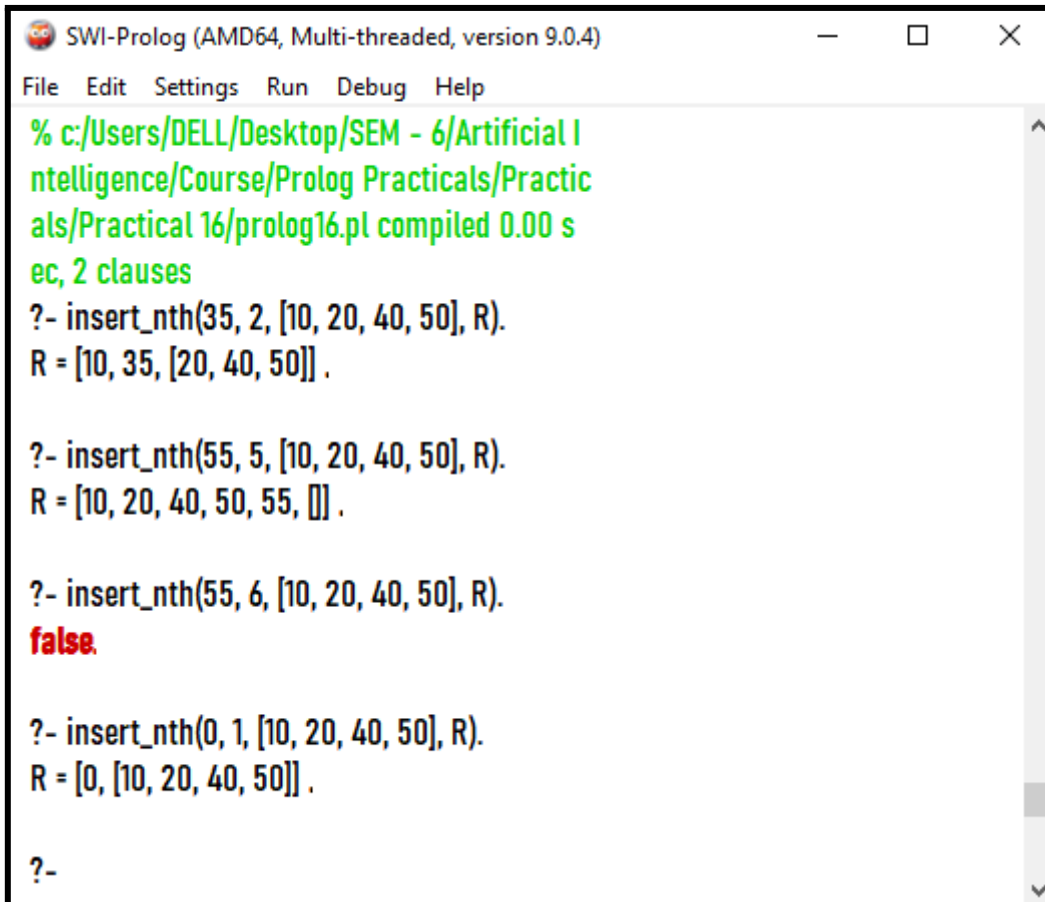


The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following content:

```
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 15/prolog15.pl compiled 0.00 sec, 0 clauses  
?- maxlist([10,20,30,52,12,100,112,80], R).  
R = 112 .  
  
?- maxlist([10,20,30,52,12,180,502], R).  
R = 502 .  
  
?- |
```

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.

```
insert_nth(I,1,List,[I,List]).
insert_nth(I,N,[H|T],[H|R]):- N1 is N-1, insert_nth(I,N1,T,R).
```



The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following content:

```
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 16/prolog16.pl compiled 0.00 sec, 2 clauses
?- insert_nth(35, 2, [10, 20, 40, 50], R).
R = [10, 35, [20, 40, 50]] .

?- insert_nth(55, 5, [10, 20, 40, 50], R).
R = [10, 20, 40, 50, 55, []] .

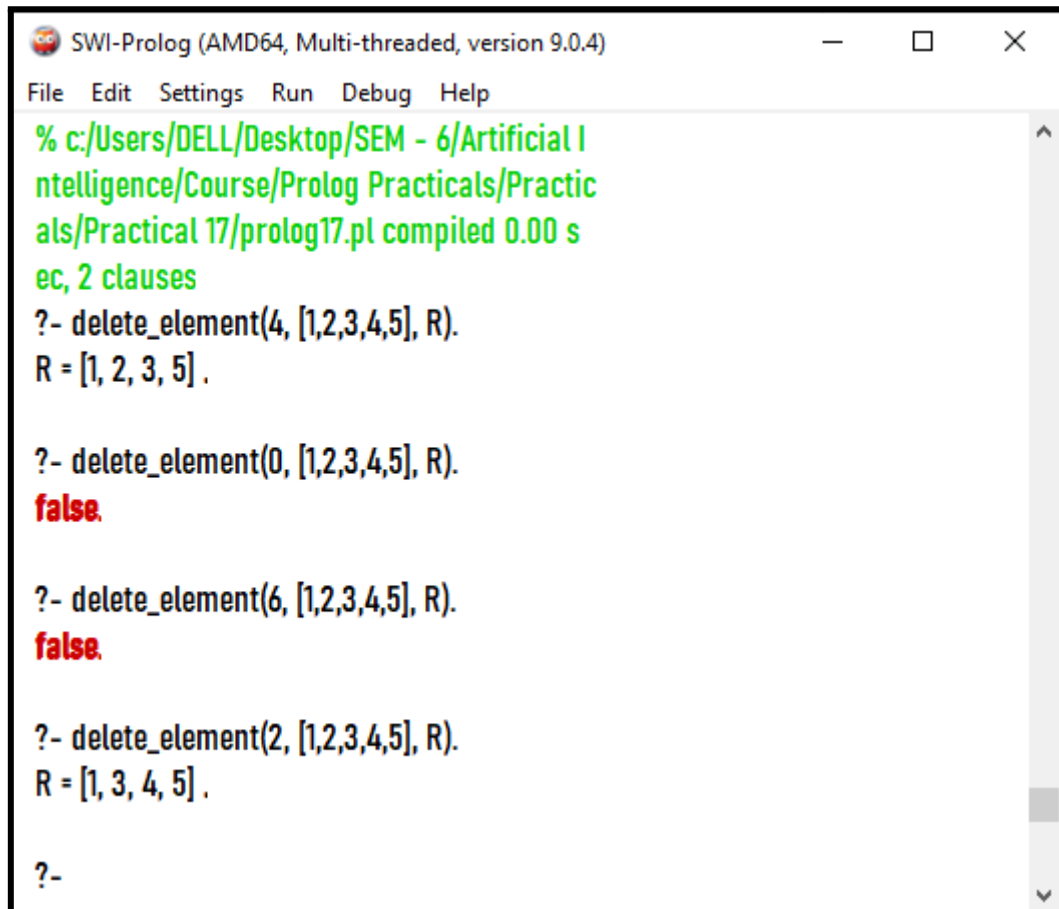
?- insert_nth(55, 6, [10, 20, 40, 50], R).
false.

?- insert_nth(0, 1, [10, 20, 40, 50], R).
R = [0, [10, 20, 40, 50]] .

?-
```

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.

```
delete_element(1,[H|T],T).  
delete_element(N,[H|T],[H|R]):- N1 is N-1, delete_element(N1,T,R).
```

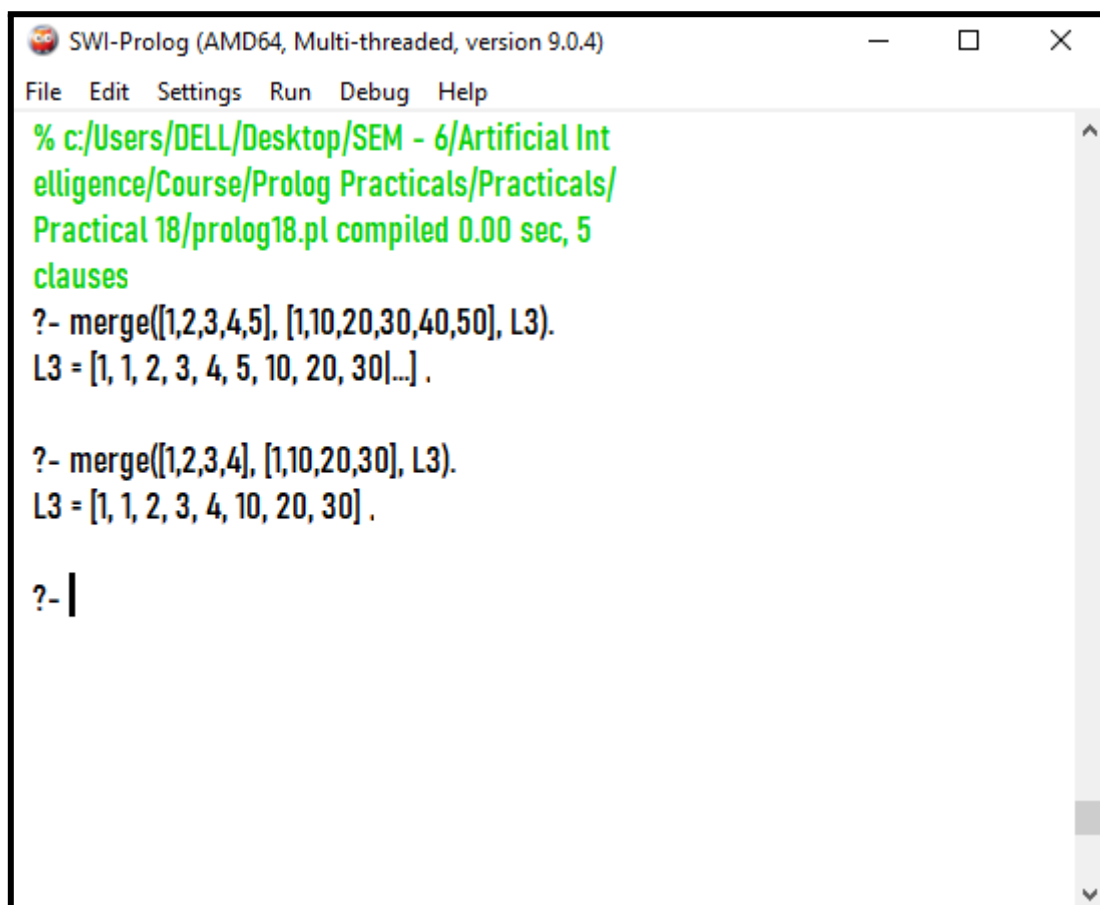


The screenshot shows the SWI-Prolog (AMD64, Multi-threaded, version 9.0.4) window. The menu bar includes File, Edit, Settings, Run, Debug, and Help. The main text area displays the following text:

```
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/Practical 17/prolog17.pl compiled 0.00 s  
ec, 2 clauses  
?- delete_element(4, [1,2,3,4,5], R).  
R = [1, 2, 3, 5].  
  
?- delete_element(0, [1,2,3,4,5], R).  
false.  
  
?- delete_element(6, [1,2,3,4,5], R).  
false.  
  
?- delete_element(2, [1,2,3,4,5], R).  
R = [1, 3, 4, 5].  
  
?-
```

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

```
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).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)  
File Edit Settings Run Debug Help  
% c:/Users/DELL/Desktop/SEM - 6/Artificial Intelligence/Course/Prolog Practicals/Practicals/  
Practical 18/prolog18.pl compiled 0.00 sec, 5  
clauses  
?- merge([1,2,3,4,5], [1,10,20,30,40,50], L3).  
L3 = [1, 1, 2, 3, 4, 5, 10, 20, 30|...] .  
  
?- merge([1,2,3,4], [1,10,20,30], L3).  
L3 = [1, 1, 2, 3, 4, 10, 20, 30] .  
  
?- |
```