

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

sum(A,B,S) :- S is A + B.

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/1.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/1.pl compiled, 0 lines read - 387 bytes written, 11 ms
| ?- sum(2,3,Z).

Z = 5

yes
| ?-
```

2. Write a Prolog program to implement 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.

max(X,Y,M) :- X<Y , M is Y.

max(X,Y,M) :- X=Y , write('both are equal').

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/2.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/2.pl compiled, 1 lines read - 670 bytes written, 11 ms
| ?- max(2,3,Z).

Z = 3

yes
| ?- max(2,1,Z).

Z = 2 ? |
```

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,!

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/3.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/3.pl compiled, 1 lines read - 834 bytes written, 10 ms
| ?- factorial(3,F).

F = 6

yes
| ?- factorial(1,F).

F = 1

yes
| ?- |
```

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

generate_fib(1,0).

generate_fib(2,1).

generate_fib(N,T):- N1 is N-1 , N2 is N-2 , generate_fib(N1,T1) , generate_fib(N2,T2) , T is T1+T2,!

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/4.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/4.pl compiled, 2 lines read - 1178 bytes written, 10 ms
| ?- generate_fib(5,F).

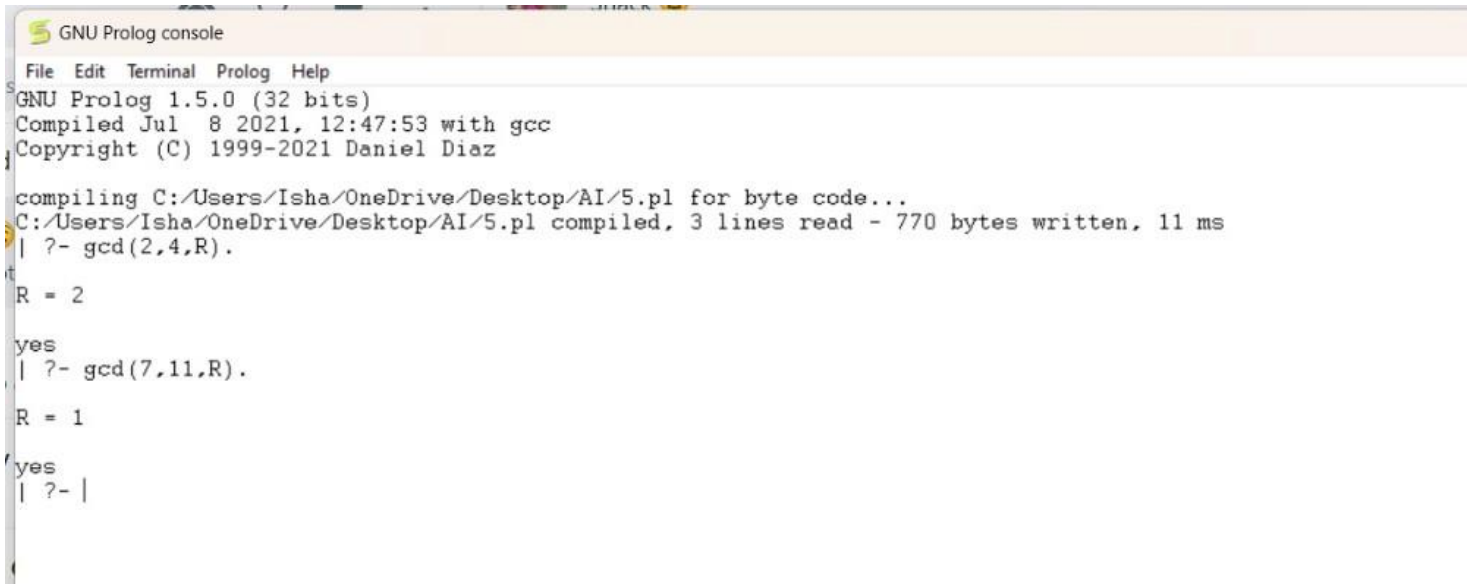
F = 3

yes
| ?- generate_fib(1,F).

F = 0 ? |
```

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

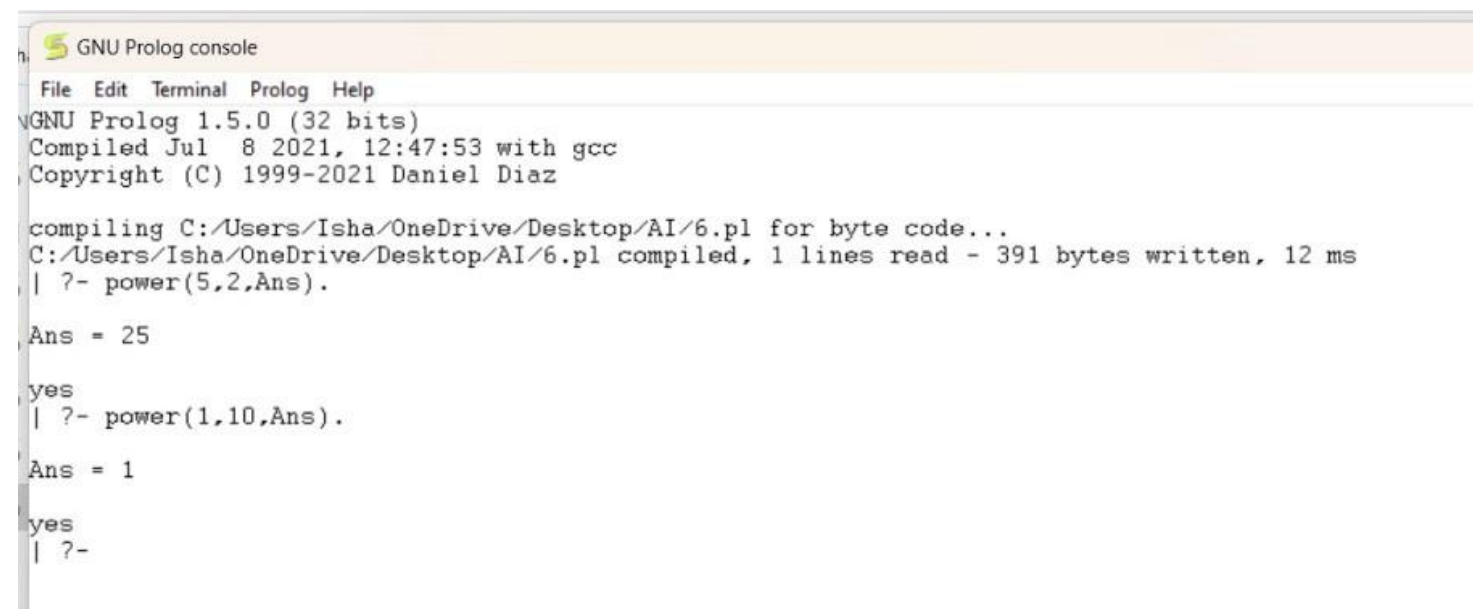
```
gcd(0,A,A) :- !.  
gcd(A,0,A) :- !.  
gcd(A,B,R) :- B1 is mod(A,B) , gcd(B,B1,R).
```



```
GNU Prolog console  
File Edit Terminal Prolog Help  
GNU Prolog 1.5.0 (32 bits)  
Compiled Jul  8 2021, 12:47:53 with gcc  
Copyright (C) 1999-2021 Daniel Diaz  
  
compiling C:/Users/Isha/OneDrive/Desktop/AI/5.pl for byte code...  
C:/Users/Isha/OneDrive/Desktop/AI/5.pl compiled, 3 lines read - 770 bytes written, 11 ms  
| ?- gcd(2,4,R).  
  
R = 2  
  
yes  
| ?- gcd(7,11,R).  
  
R = 1  
  
yes  
| ?- |
```

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

```
power(Num,Pow,Ans) :- Ans is Num^Pow.
```



```
GNU Prolog console  
File Edit Terminal Prolog Help  
GNU Prolog 1.5.0 (32 bits)  
Compiled Jul  8 2021, 12:47:53 with gcc  
Copyright (C) 1999-2021 Daniel Diaz  
  
compiling C:/Users/Isha/OneDrive/Desktop/AI/6.pl for byte code...  
C:/Users/Isha/OneDrive/Desktop/AI/6.pl compiled, 1 lines read - 391 bytes written, 12 ms  
| ?- power(5,2,Ans).  
  
Ans = 25  
  
yes  
| ?- power(1,10,Ans).  
  
Ans = 1  
  
yes  
| ?-
```

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

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

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/7.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/7.pl compiled, 1 lines read - 391 bytes written, 9 ms
| ?- multi(3,6,R).

R = 18

yes
| ?- multi(10,8,R).

R = 80

yes
| ?-
```

8. Write a program in PROLOG to implement towerofhanoi (N) where N represents the number ofdiscs

```
move(1,X,Y,_):-
    write('Move top disk from '), write(X), write(' to '), write(Y), nl.
move(N,X,Y,Z):-
    N>1,
    M is N-1,
    move(M,X,Z,Y),
    move(1,X,Y,_),
    move(M,Z,Y,X).
```

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/8.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/8.pl compiled, 7 lines read - 1401 bytes written, 10 ms
| ?- move(4,source,target,auxiliary).
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
Move top disk from source to auxiliary
Move top disk from target to source
Move top disk from target to auxiliary
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
Move top disk from auxiliary to source
Move top disk from target to source
Move top disk from auxiliary to target
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
true ? |
```

9. Consider a cyclic directed graph [edge (p, q), edge (q, r), edge (q, r), edge (q, s), edge (s,t)] where edge (A,B) is a predicate indicating directed edge in a graph from a node A to a node B. Write a program to check whether there is a route from one node to another node.

```
edge(p, q).
edge(q, r).
edge(q, s).
edge(s, t).
edge(q, r).
```

```
route(Node1, Node2) :-
    dfs(Node1, Node2, [Node1]).
```

```
dfs(Node, Node, _) :- !.
dfs(Node1, Node2, Visited) :-
    edge(Node1, X),
    not(member(X, Visited)),
    dfs(X, Node2, [X|Visited]).
```

10. Write a Prolog program to implement `memb(X, L)`: to check whether `X` is a member of `L`

or not

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

```
GNU Prolog console  
File Edit Terminal Prolog Help  
GNU Prolog 1.5.0 (32 bits)  
Compiled Jul  8 2021, 12:47:53 with gcc  
Copyright (C) 1999-2021 Daniel Diaz  
  
compiling C:/Users/Isha/OneDrive/Desktop/AI/10.pl for byte code...  
C:/Users/Isha/OneDrive/Desktop/AI/10.pl compiled, 2 lines read - 430 bytes written, 10 ms  
| ?- memb(3,[1,2,3,6,7]).  
  
true ?  
  
yes  
| ?- memb(8,[1,2,3,6,7]).  
  
no  
| ?-
```

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

```
File Edit Terminal Prolog Help  
GNU Prolog 1.5.0 (32 bits)  
Compiled Jul  8 2021, 12:47:53 with gcc  
Copyright (C) 1999-2021 Daniel Diaz  
  
compiling C:/Users/Isha/OneDrive/Desktop/AI/11.pl for byte code...  
C:/Users/Isha/OneDrive/Desktop/AI/11.pl compiled, 3 lines read - 484 bytes written, 9 ms  
| ?- conc([1,2], [3,4], L3).  
  
L3 = [1,2,3,4]  
  
yes  
| ?- conc([1,2,9], [3,4], L3).  
  
L3 = [1,2,9,3,4]  
  
yes  
| ?-
```

12. Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversedlist.

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

```
GNU Prolog console  
File Edit Terminal Prolog Help  
GNU Prolog 1.5.0 (32 bits)  
Compiled Jul  8 2021, 12:47:53 with gcc  
Copyright (C) 1999-2021 Daniel Diaz  
  
compiling C:/Users/Isha/OneDrive/Desktop/AI/12.pl for byte code...  
C:/Users/Isha/OneDrive/Desktop/AI/12.pl compiled, 4 lines read - 656 bytes written, 15 ms  
error: C:/Users/Isha/OneDrive/Desktop/AI/12.pl:1: native code procedure reverse/2 cannot be redefined (ignored)  
| ?- reverse([1,2,3,4], R).  
  
R = [4,3,2,1]  
  
yes  
| ?- reverse([0,4,6,3,8], R).  
  
R = [8,3,6,4,0]  
  
yes  
| ?-
```

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

```
palind([]):- write('palindrome').  
palind([_]):- write('palindrome').  
palind(L) :-  
    append([H|T], [H], L),  
    palind(T)  
;  
write('Not a palindrome').
```

```
GNU Prolog console  
File Edit Terminal Prolog Help  
GNU Prolog 1.5.0 (32 bits)  
Compiled Jul  8 2021, 12:47:53 with gcc  
Copyright (C) 1999-2021 Daniel Diaz  
  
compiling C:/Users/Isha/OneDrive/Desktop/AI/13.pl for byte code...  
C:/Users/Isha/OneDrive/Desktop/AI/13.pl compiled, 9 lines read - 1128 bytes written, 12 ms  
| ?- palind([a,n,s,h]).  
Not a palindrome  
  
yes  
| ?- palind([a,n,n,a]).  
palindrome  
  
true ? |
```

14. 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, Rest), S is H + Rest.`

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/14.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/14.pl compiled, 5 lines read - 643 bytes written, 12 ms
| ?- sumlist([1,2,3,4], S).

S = 10

yes
| ?- sumlist([5,-2,3,4], S).

S = 10

yes
| ?-
```

15. 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

`evenlength([]).`

`evenlength([_|T]) :- oddlength(T).`

`oddlength([_|_]).`

`oddlength([_|T]) :- evenlength(T).`


```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/15.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/15.pl compiled, 8 lines read - 698 bytes written, 9 ms
| ?- evenlength([1,2,3,4]).

true ?

yes
| ?- evenlength([1,2,3]).

no
| ?-

odddlength([1,2,3]).

true ?

yes
| ?-
```

16. 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`.

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

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/16.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/16.pl compiled, 3 lines read - 869 bytes written, 14 ms
| ?- nth_element(3, [1,2,4,6], X).

X = 4 ?

yes
| ?- nth_element(5, [1,2,4,6,9,10], X).

X = 9 ?

yes
| ?- |
```

17. Write a program in PROLOG to implement remove_dup (L, R) where L denotes the list with some duplicates and the list R denotes the list with duplicates removed.

```
member(X,[X|_]).
member(X,[_|Y]):-
member(X,Y).
remove_dup(L,M):
-dupacc(L,[],M).
dupacc([],A,A).
dupacc([H|T],A,L):
-
member(H,A),dup
acc(T,A,L),!.
dupacc([H|T],A,L):
-
dupacc(T,[H|A],L).
```

```

File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/17.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/17.pl compiled, 5 lines read - 1420 bytes written, 11 ms
error: C:/Users/Isha/OneDrive/Desktop/AI/17.pl:1: native code procedure member/2 cannot be redefined (ignored)
| ?- remove_dup([1,3,3,5,6], R).

R = [6,5,3,1]

yes
| ?- remove_dup([1,2,2,3,5,6], R).

R = [6,5,3,2,1]

yes
| ?-

```

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

```

maxlist([X], X).
maxlist([H|T], M) :-
    maxlist(T, M1),
    M is max(H, M1).

```

```

File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/18.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/18.pl compiled, 1 lines read - 677 bytes written, 12 ms
| ?- maxlist([1,8,5,3,2], M).

M = 8 ?

yes
| ?- maxlist([1,-9,5,3,2], M).

M = 5 ?

yes
| ?- |

```

19. Write a prolog program to implement insert_nth(I, N, L, R) that inserts an item I into Nthposition of list L to generate a list R.

```

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).

```

```

GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/19.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/19.pl compiled, 4 lines read - 983 bytes written, 9 ms
| ?- insert_nth(5,3,[1,2,6,7,4],R).

R = [1,2,5,6,7,4] ?

```

20. Write a Program in PROLOG to implement sublist(S, L) that checks whether the list S is the sublist of list L or not. (Check for sequence or the part in the same order).

```

sublist([], _).
sublist([H|T], L) :-
    append(_, [H|L2], L),
    sublist_helper(T, L2).

```

```

sublist_helper([], _).
sublist_helper([H|T], [H|L2]) :-
    sublist_helper(T, L2).

```

```

GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/20.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/20.pl compiled, 5 lines read - 1015 bytes written, 15 ms
error: C:/Users/Isha/OneDrive/Desktop/AI/20.pl:1: native code procedure sublist/2 cannot be redefined (ignored
| ?- sublist([1,6],[1,2,5,6])).

true ?

yes
| ?- sublist([1,6],[1,2,5]).

no
| ?-

```

21. 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_nth(1, [_|T], T).
delete_nth(N, [H|T], [H|R]) :-
    N > 1,
    N1 is N-1,
    delete_nth(N1, T, R).

```

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/21.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/21.pl compiled, 4 lines read - 918 bytes written, 16 ms
| ?- delete_nth(3,[1,2,9,3,4,5],R).

R = [1,2,3,4,5] ?
yes
| ?-
```

22. Write a program in PROLOG to implement delete_all (X, L, R) where X denotes the element whose all occurrences has to be deleted from list L to obtain list R.

```
delete_all(_, [], []).
delete_all(X, [X|T], R) :-
    delete_all(X, T, R).
delete_all(X, [H|T], [H|R]) :-
    X \= H,
    delete_all(X, T, R).
```

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/22.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/22.pl compiled, 4 lines read - 851 bytes written, 12 ms
| ?- delete_all(3,[3,1,5,6,3,2,3,3],R).

R = [1,5,6,2] ?
```

23. 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([], L, L).
merge(L, [], L).
merge([H1|T1], [H2|T2], [H1|T]) :-
    H1 <= H2,
    merge(T1, [H2|T2], T).
merge([H1|T1], [H2|T2], [H2|T]) :-
    H1 > H2,
    merge([H1|T1], T2, T).
```

```
GNU Prolog console
File Edit Terminal Prolog Help
GNU Prolog 1.5.0 (32 bits)
Compiled Jul  8 2021, 12:47:53 with gcc
Copyright (C) 1999-2021 Daniel Diaz

compiling C:/Users/Isha/OneDrive/Desktop/AI/23.pl for byte code...
C:/Users/Isha/OneDrive/Desktop/AI/23.pl compiled, 5 lines read - 1340 bytes written, 7 ms
| ?- merge([1,2,3],[8,9,10],R).

R = [1,2,3,8,9,10] ? |
```

24. Write a PROLOG program that will take grammar rules in the following

format:NT -> (NT | T)*

Where NT is any nonterminal, T is any terminal and Kleene star (*) signifies any number of repetitions, and generate the corresponding top-down parser, that is:

sentence -> noun-phrase, verb-

phrasedeterminer -> [the]

will generate the following:

sentence(I, O) :- noun_phrase(I,R), verb_phrase(R,O).

determiner([the|X], X) :- !.

sentence(I, O) :- noun_phrase(I, R), verb_phrase(R, O).

noun_phrase(I, O) :- article(I, R), noun(R, O).

noun_phrase(I, O) :- noun(I, O).

verb_phrase(I, O) :- verb(I, R), noun_phrase(R, O).

verb_phrase(I, O) :- verb(I, O).

article([the|X], X) :- !.

noun([dog|X], X) :- !.

noun([cat|X], X) :- !.

verb([chases|X], X) :- !.

verb([sees|X], X) :- !.

25. Write a prolog program that implements Semantic Networks

(ATN/RTN).

node(lion).

node(tiger).

node(animal).

node(feline).

rel(is_a, lion, animal).

rel(is_a, tiger, animal).

rel(is_a, feline, animal).

rel(is_a, lion, feline).
rel(is_a, tiger, feline).

?- rel(is_a, lion, animal).
Yes

?- rel(is_a, X, feline).
X = lion ;
X = tiger.

26. Write a prolog program that implements Semantic Networks (ATN/RTN).

```
node(lion).  
node(tiger).  
node(animal).  
node(feline).
```

```
rel(is_a, lion, animal).  
rel(is_a, tiger, animal).  
rel(is_a, feline, animal).  
rel(is_a, lion, feline).  
rel(is_a, tiger, feline).
```

```
?- rel(is_a, lion, animal).  
Yes
```

```
?- rel(is_a, X, feline).  
X = lion ;  
X = tiger.
```