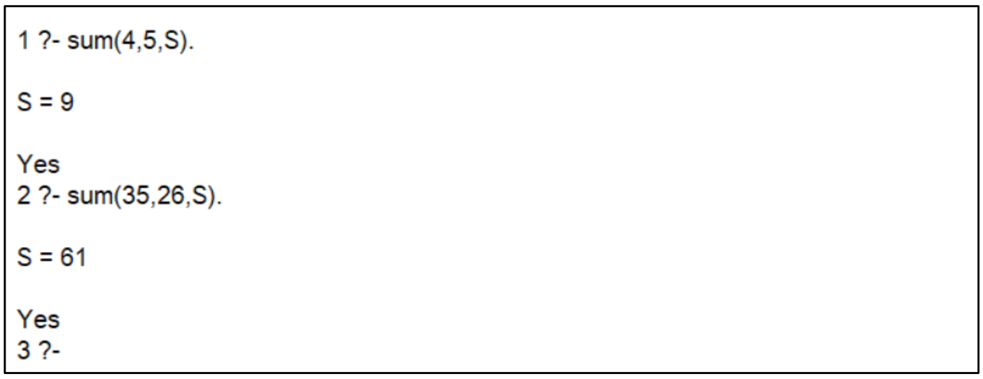
Harsh (20058570011)

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

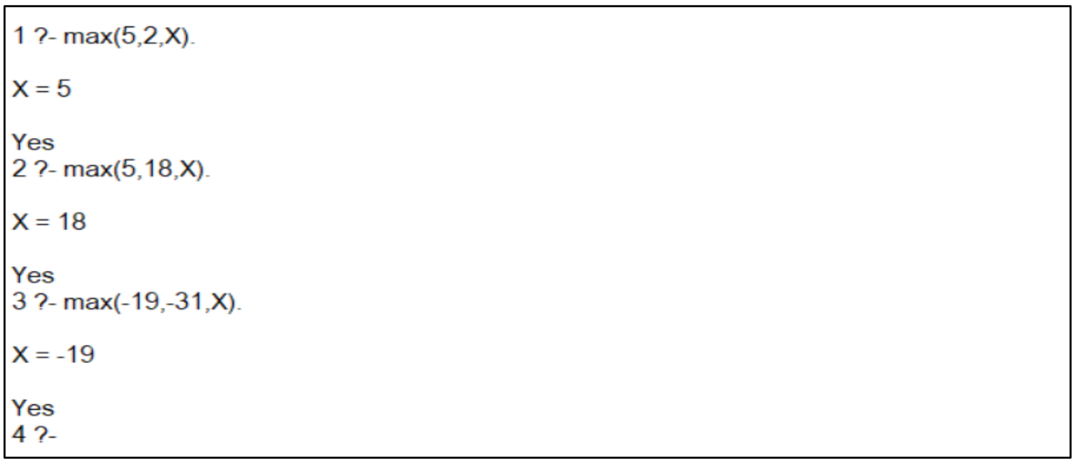
```prolog
sum(X,Y,Z) :-
   Z is X+Y.
```



```
1 ?- sum(4,5,S).

S = 9

Yes
2 ?- sum(35,26,S).

S = 61

Yes
3 ?-
```

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

```prolog
max(X,Y,M) :- X > Y, M is X.
max(X,Y,M) :- X =< Y, M is Y.
```

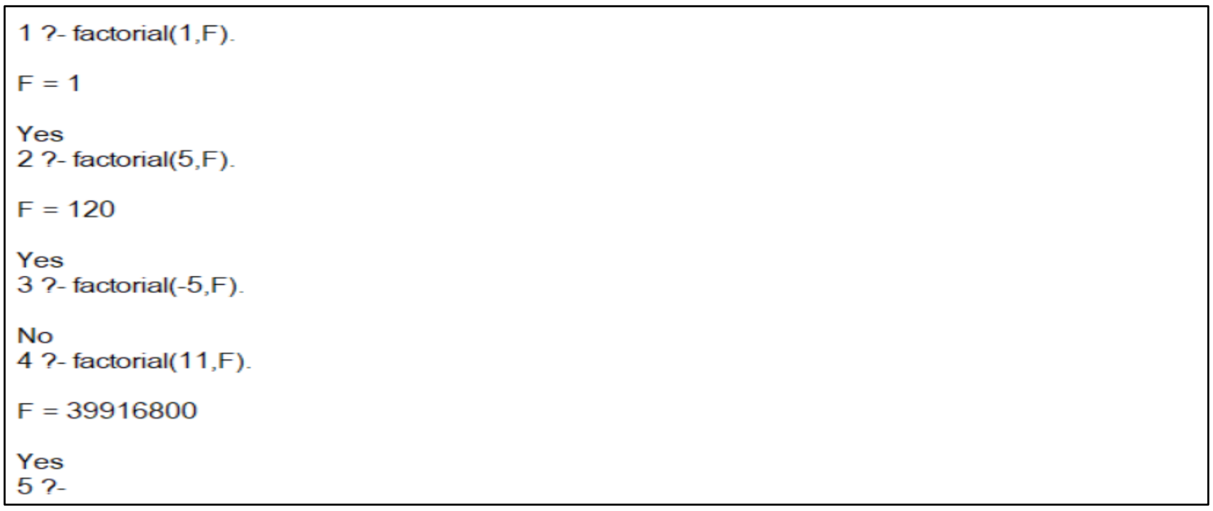


```
1 ?- max(5,2,X).

X = 5

Yes
2 ?- max(5,18,X).

X = 18

Yes
3 ?- max(-19,-31,X).

X = -19

Yes
4 ?-
```

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

```prolog
factorial(0,1).
factorial(N,F) :-
    N > 0,
    N1 is N - 1,
    factorial(N1,F1),
    F is N * F1.
```

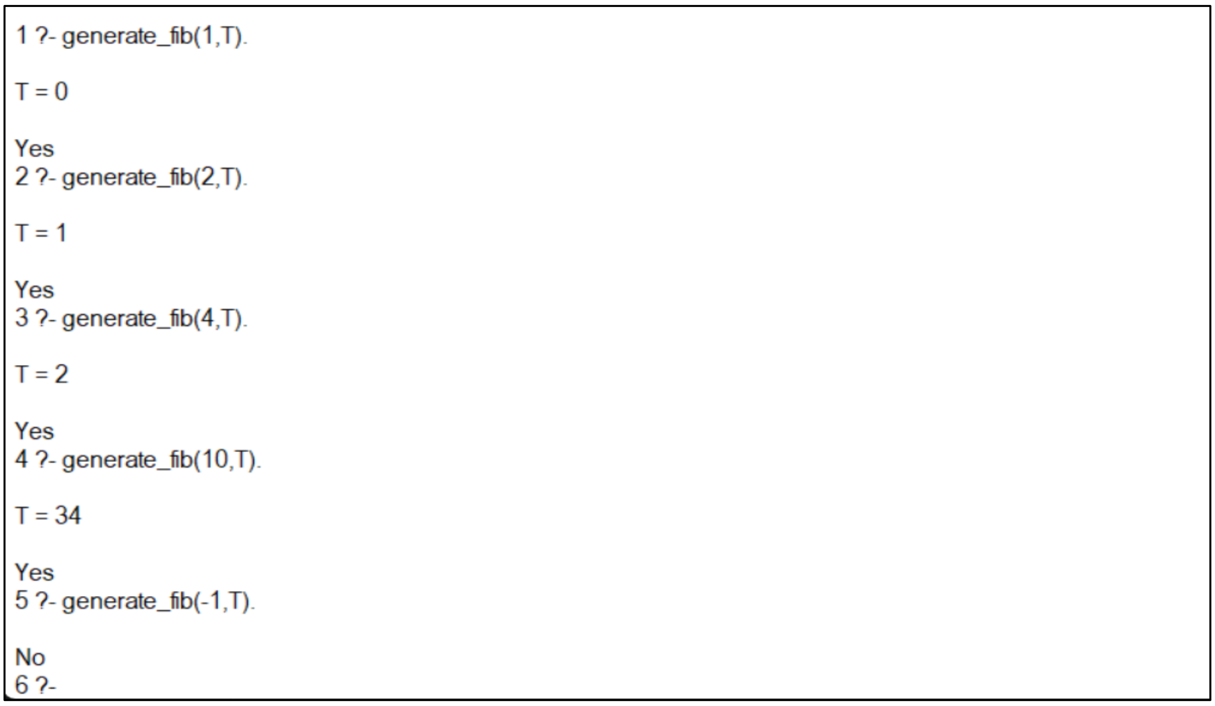
Harsh (20058570011)



```
1 ?- factorial(1,F).

F = 1

Yes
2 ?- factorial(5,F).

F = 120

Yes
3 ?- factorial(-5,F).

No
4 ?- factorial(11,F).

F = 39916800

Yes
5 ?-
```

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

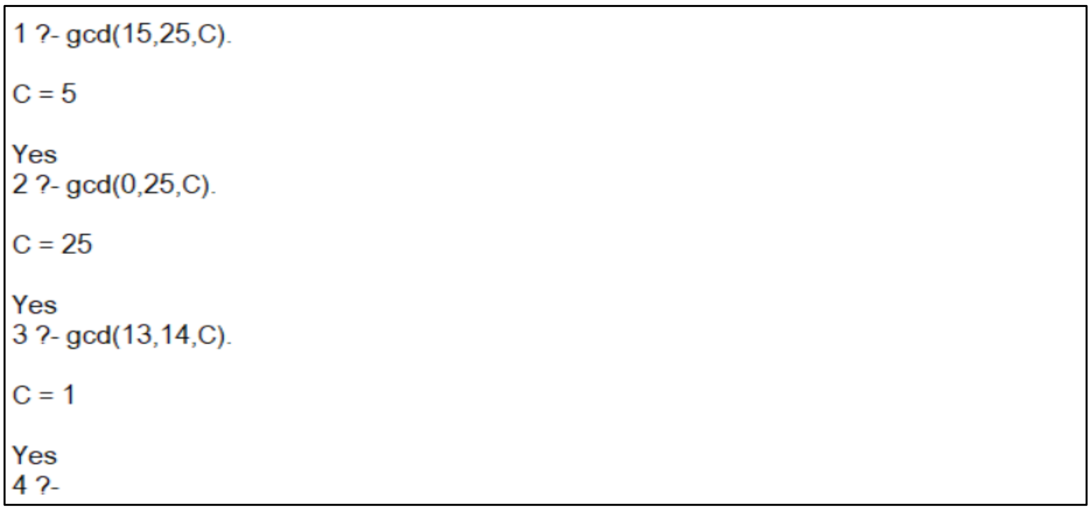


```
1 ?- generate_fib(1,T).

T = 0

Yes
2 ?- generate_fib(2,T).

T = 1

Yes
3 ?- generate_fib(4,T).

T = 2

Yes
4 ?- generate_fib(10,T).

T = 34

Yes
5 ?- generate_fib(-1,T).

No
6 ?-
```

Harsh (20058570011)

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

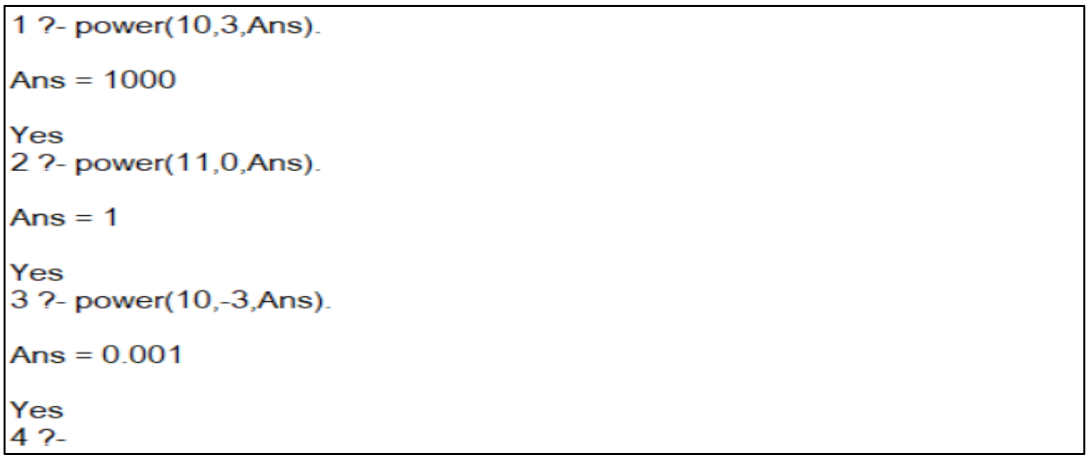
```
gcd(X,X,X).
gcd(0,X,X).
gcd(X,0,X).
gcd(X,Y,G) :-
    X > Y,
    X1 is X - Y,
    gcd(X1,Y,G).
gcd(X,Y,G) :-
    X < Y,
    Y1 is Y - X,
    gcd(X,Y1,G).
```



```
1 ?- gcd(15,25,C).

C = 5

Yes
2 ?- gcd(0,25,C).

C = 25

Yes
3 ?- gcd(13,14,C).

C = 1

Yes
4 ?-
```

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

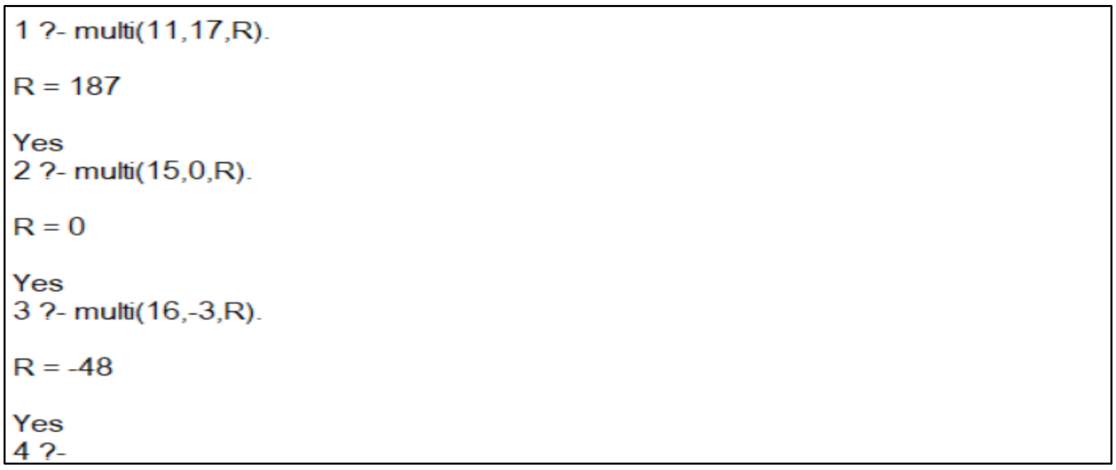
```
power(Num,0,1).
power(Num,Pow,Ans) :-
    Pow > 0,
    Pow1 is Pow - 1,
    power(Num,Pow1,Ans1),
    Ans is Num * Ans1.
power(Num,Pow,Ans) :-
    Pow < 0,
    Pow1 is Pow + 1,
    power(Num,Pow1,Ans1),
    Ans is Ans1 / Num.
```



```
1 ?- power(10,3,Ans).

Ans = 1000

Yes
2 ?- power(11,0,Ans).

Ans = 1

Yes
3 ?- power(10,-3,Ans).

Ans = 0.001

Yes
4 ?-
```

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

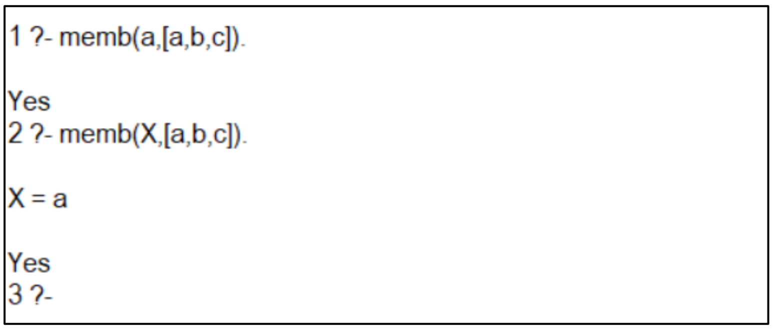
```
multi(N1, N2, R) :-
    R is N1 * N2.
```



```
1 ?- multi(11,17,R).

R = 187

Yes
2 ?- multi(15,0,R).

R = 0

Yes
3 ?- multi(16,-3,R).

R = -48

Yes
4 ?-
```

8. 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).
```

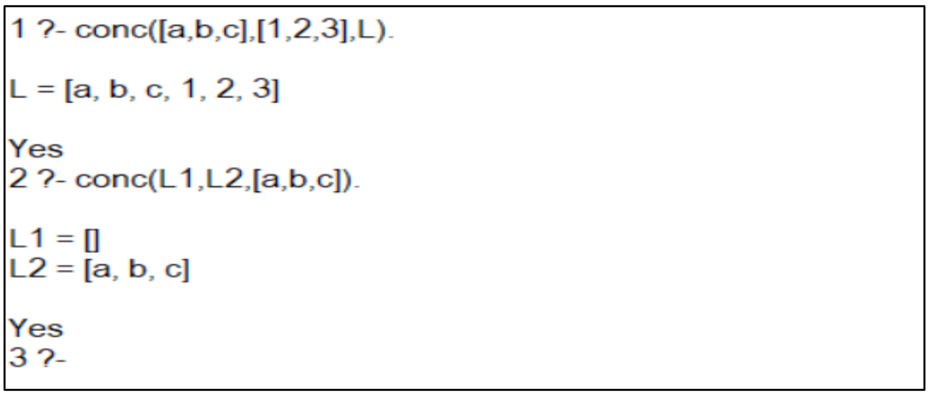


```
1 ?- memb(a,[a,b,c]).

Yes
2 ?- memb(X,[a,b,c]).

X = a

Yes
3 ?-
```

Harsh (20058570011)

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.

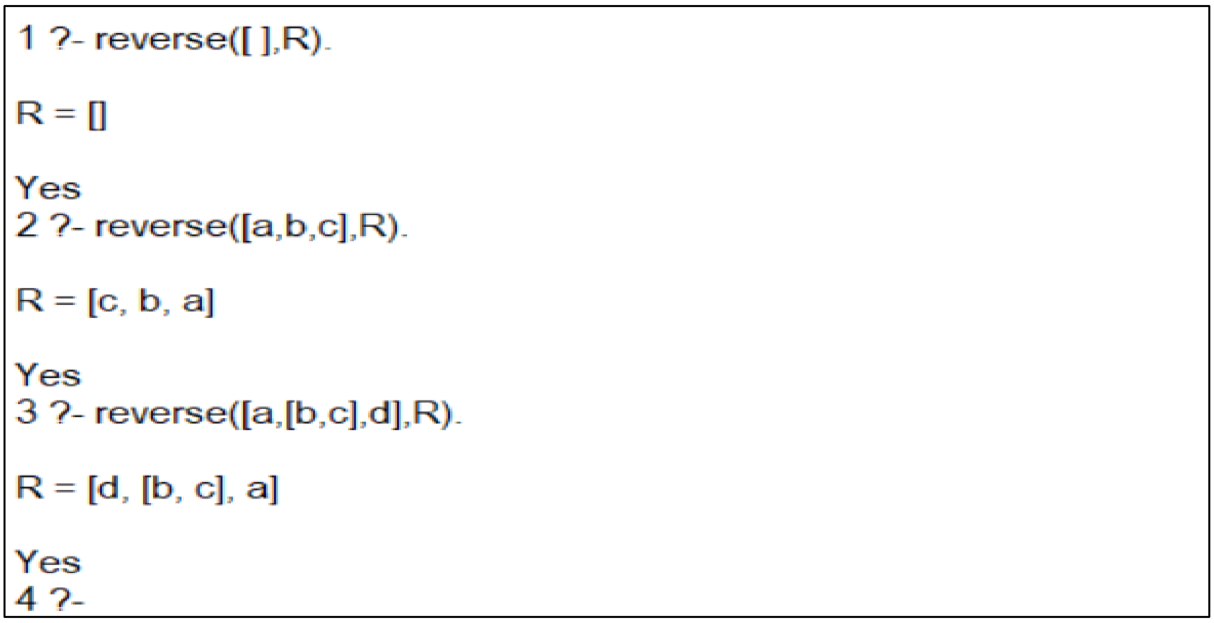
```
conc([], L, L).
conc([H|T], L, [H|L1]) :-
    conc(T, L, L1).
```



```
1 ?- conc([a,b,c],[1,2,3],L).

L = [a, b, c, 1, 2, 3]

Yes
2 ?- conc(L1,L2,[a,b,c]).

L1 = []
L2 = [a, b, c]

Yes
3 ?-
```

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

```
conc([], L, L).
conc([H|T], L, [H|L1]) :-
    conc(T, L, L1).

reverse([], []).
reverse([H|T], R) :-
    reverse(T, R1),
    conc(R1, [H], R).
```

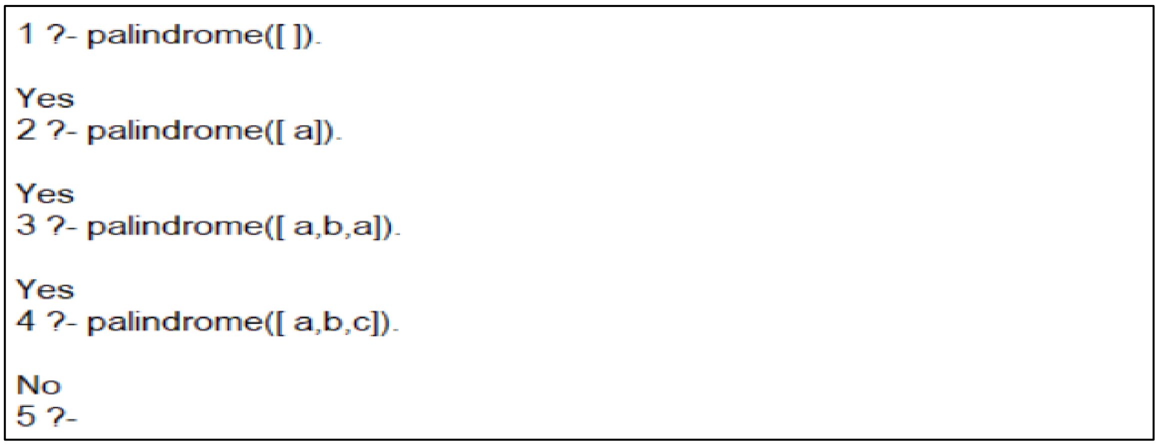


```
1 ?- reverse([ ],R).

R = []

Yes
2 ?- reverse([a,b,c],R).

R = [c, b, a]

Yes
3 ?- reverse([a,[b,c],d],R).

R = [d, [b, c], a]

Yes
4 ?-
```

Harsh (20058570011)

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

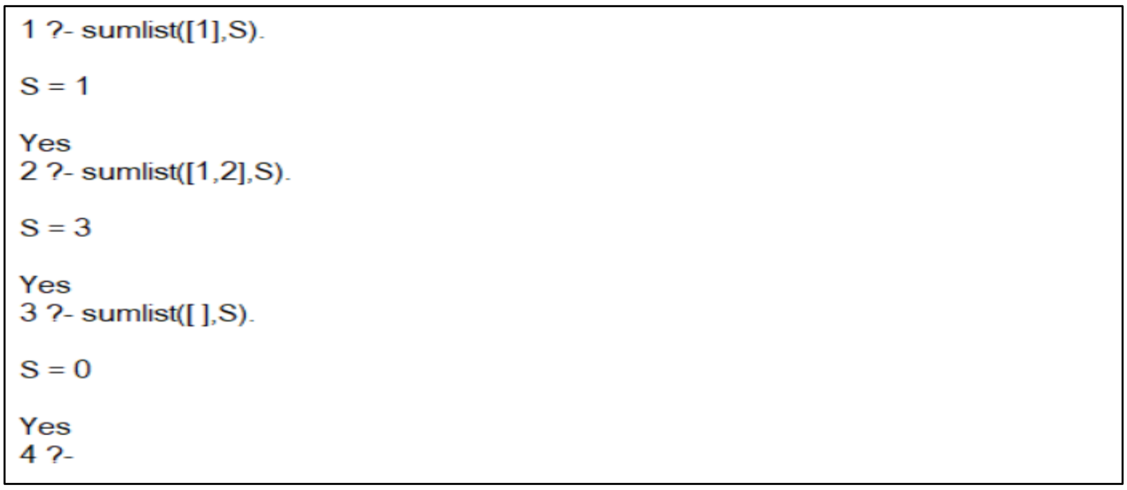
```
conc([], L, L).
conc([H|T], L, [H|L1]) :-
    conc(T, L, L1).

reverse([], []).
reverse([H|T], R) :-
    reverse(T, R1),
    conc(R1, [H], R).

palindrome(L) :-
    reverse(L, L).
```



```
1 ?- palindrome([ ]).

Yes
2 ?- palindrome([ a]).

Yes
3 ?- palindrome([ a,b,a]).

Yes
4 ?- palindrome([ a,b,c]).

No
5 ?-
```

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 S1 + H.
```

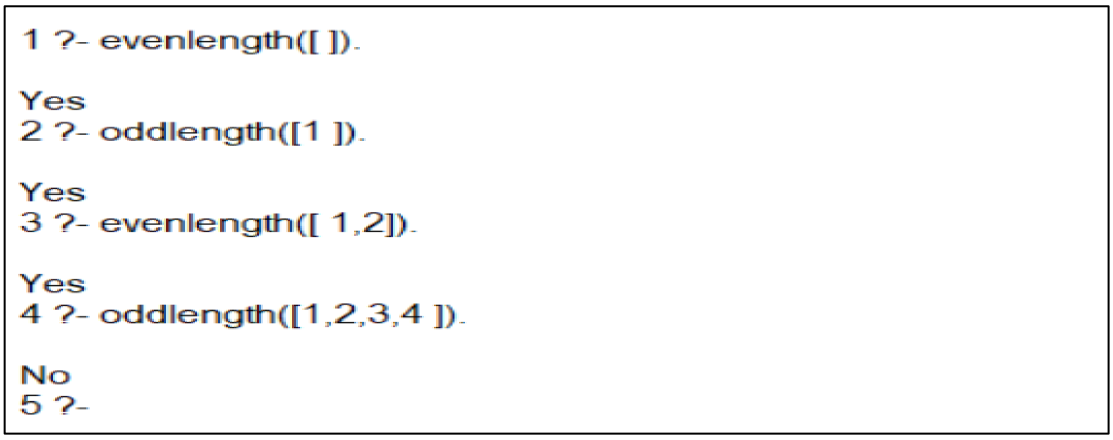


```
1 ?- sumlist([1],S).

S = 1

Yes
2 ?- sumlist([1,2],S).

S = 3

Yes
3 ?- sumlist([ ],S).

S = 0

Yes
4 ?-
```

Harsh (20058570011)

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.

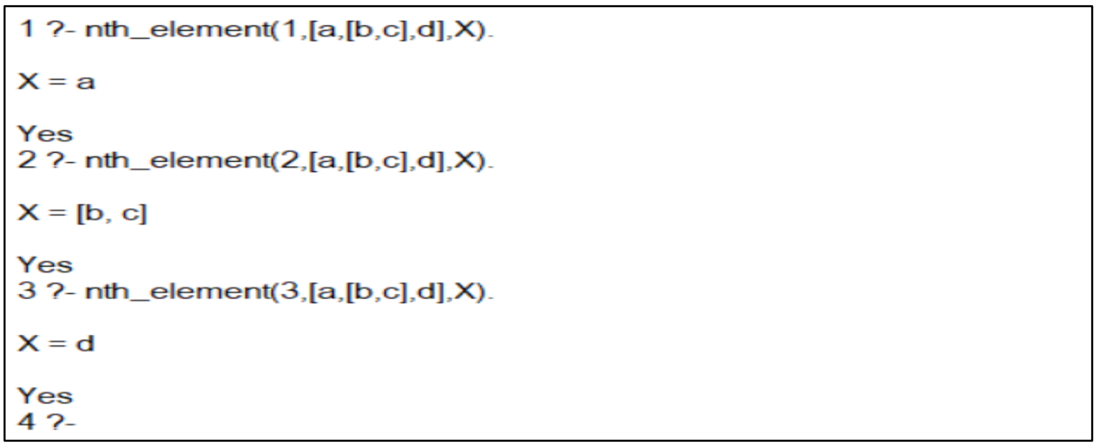
```
evenlength([]).
evenlength([_|T]) :-
    oddlength(T).

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



```
1 ?- evenlength([ ]).

Yes
2 ?- oddlength([1 ]).

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

Yes
4 ?- oddlength([1,2,3,4 ]).

No
5 ?-
```

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.

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

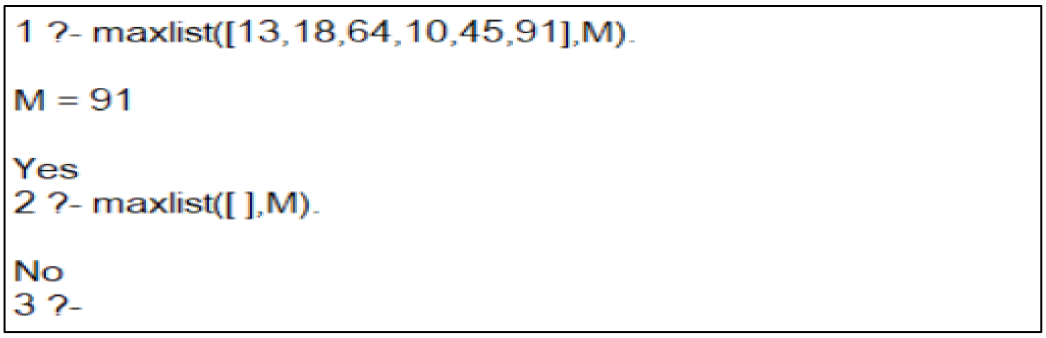


```
1 ?- nth_element(1,[a,[b,c],d],X).

X = a

Yes
2 ?- nth_element(2,[a,[b,c],d],X).

X = [b, c]

Yes
3 ?- nth_element(3,[a,[b,c],d],X).

X = d

Yes
4 ?-
```

Harsh (20058570011)

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

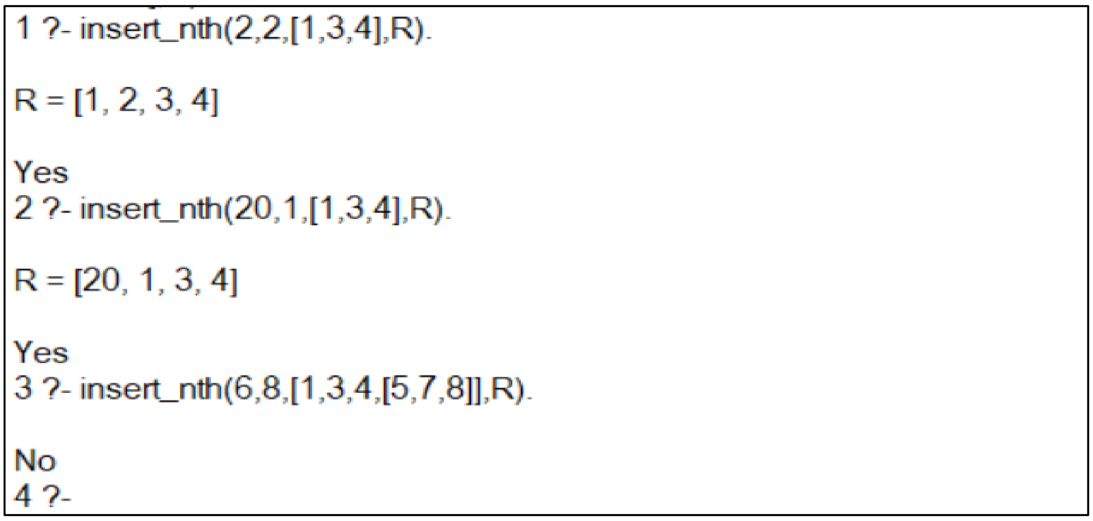
```
max(X,Y,M) :- X > Y, M is X.
max(X,Y,M) :- X =< Y, M is Y.

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



```
1 ?- maxlist([13,18,64,10,45,91],M).

M = 91

Yes
2 ?- maxlist([ ],M).

No
3 ?-
```

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

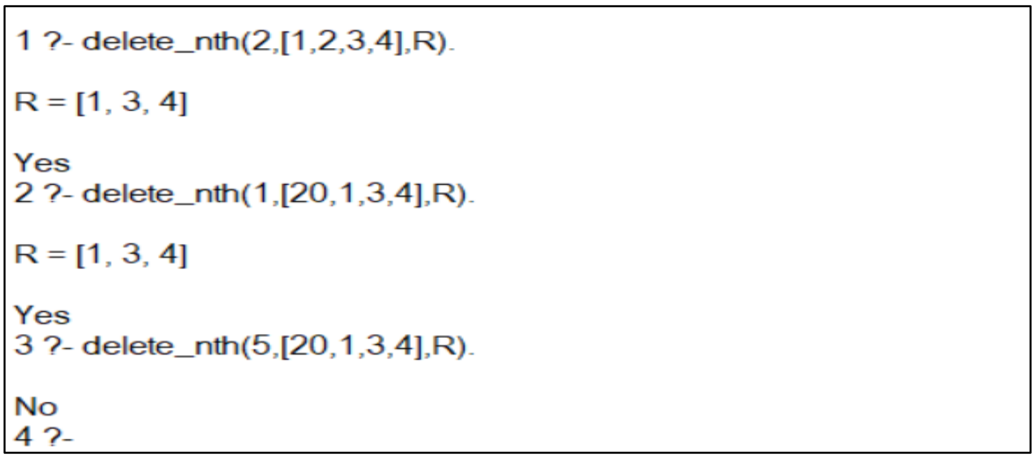


```
1 ?- insert_nth(2,2,[1,3,4],R).

R = [1, 2, 3, 4]

Yes
2 ?- insert_nth(20,1,[1,3,4],R).

R = [20, 1, 3, 4]

Yes
3 ?- insert_nth(6,8,[1,3,4,[5,7,8]],R).

No
4 ?-
```

Harsh (20058570011)

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.

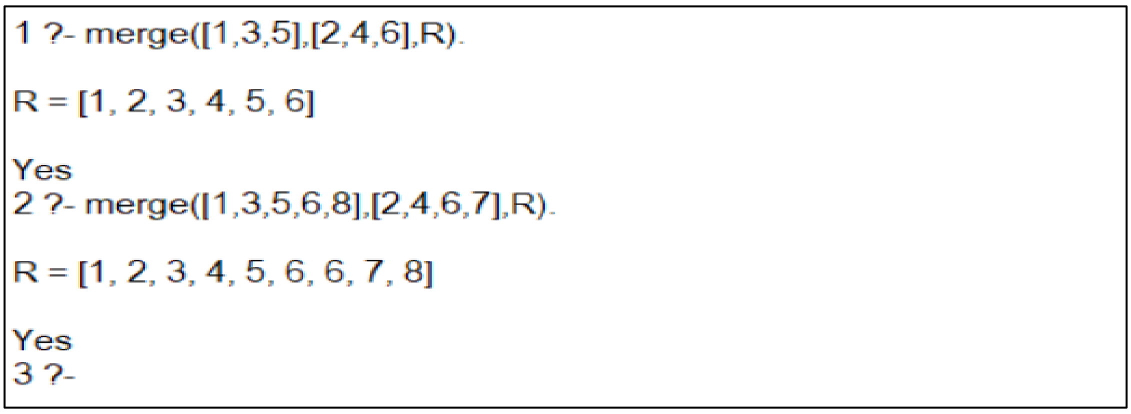
```
delete_nth(1, [_|T], T).
delete_nth(N, [H|T], [H|R]) :-
     N > 1,
     N1 is N-1,
    delete_nth(N1, T, R).
```



```
1 ?- delete_nth(2,[1,2,3,4],R).

R = [1, 3, 4]

Yes
2 ?- delete_nth(1,[20,1,3,4],R).

R = [1, 3, 4]

Yes
3 ?- delete_nth(5,[20,1,3,4],R).

No
4 ?-
```

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([],L,L).
merge(L,[],L).
merge([H1|T1],[H2|T2],[H1|T3]) :-
   H1 =< H2,
   merge(T1,[H2|T2],T3).
merge([H1|T1],[H2|T2],[H2|T3]) :-
   H1 > H2,
   merge([H1|T1],T2,T3).
```



```
1 ?- merge([1,3,5],[2,4,6],R).

R = [1, 2, 3, 4, 5, 6]

Yes
2 ?- merge([1,3,5,6,8],[2,4,6,7],R).

R = [1, 2, 3, 4, 5, 6, 6, 7, 8]

Yes
3 ?-
```