DEEN DAYAL UPADHYAYA COLLEGE

UNIVERSITY OF DELHI



**NAME: HARSHITA SYUNARY**

**ROLL NO.: 21HCS4141**

**SEMESTER: VI**

**SUBJECT: ARTIFICIAL INTELLIGENCE**

**SUBMITTED TO: PROF. ANUJA SONI**

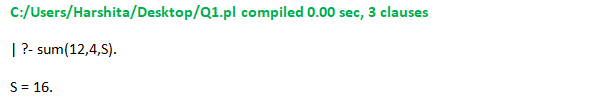
**-------------------------------------------------------------------------**

Q1. Write a prolog program to calculate the sum of two numbers.

**CODE**:

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

**OUTPUT**:



**--------------------------------------------------------------------------------------------------------------------------------------**

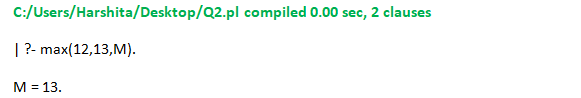
Q2. Write a Prolog program to implement max(X, Y, M) so that M is the maximum of two numbers X and Y.

**CODE**:

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

max(\_,Y,M):- M is Y.

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q3. Write a program in PROLOG to implement factorial (N, F) where F represents the factorial of a number N.

**CODE**:

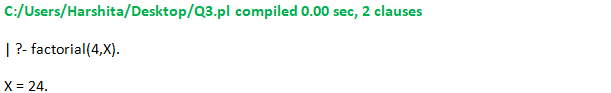
factorial(0,1).

factorial(N,X):- N1 is N-1,

factorial(N1,X1),

X is X1\*N.

**OUTPUT:**

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q4. Write a program in PROLOG to implement generate\_fib(N,T) where T represents the Nth term of the fibonacci series.

**CODE**:

generate\_fib(0,1).

generate\_fib(1,1).

generate\_fib(N,T):- N1 is N-1,

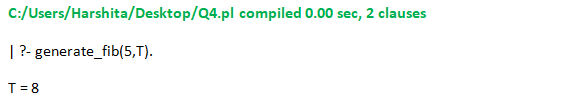
generate\_fib(N1,T1),

N2 is N-2,

generate\_fib(N2,T2),

T is T1+T2.

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q5. Write a Prolog program to implement GCD of two numbers.

**CODE:**

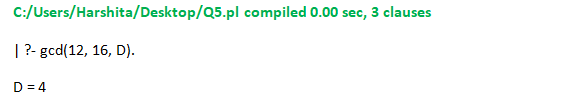
gcd(M,0,M):-!.

gcd(M,N,D):-N > 0,

X is mod(M,N),

gcd(N,X,D).

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q6. Write a Prolog program to implement power (Num,Pow, Ans) : where Num is raised to the power Pow to get Ans.

**CODE**:

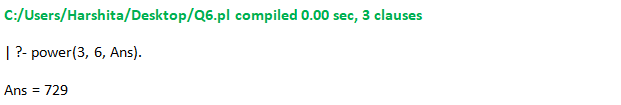
power(Num,1,Num).

power(Num,Pow,Ans):- Pow1 is Pow-1,

power(Num,Pow1,Ans1),

Ans is Ans1\*Num.

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q7.Prolog program to implement multi (N1, N2, R) : where N1 and N2 denotes the numbers to be multiplied and R represents the result.

**CODE**:

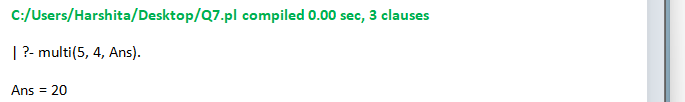
multi(N1,1,N1).

multi(N1,N2,Ans):- Temp is N2-1,

multi(N1,Temp,Ans1),

Ans is Ans1+N1.

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

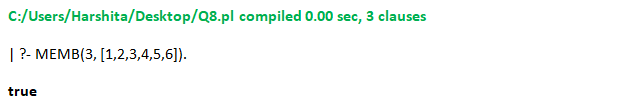
Q8. Write a Prolog program to implement memb(X, L): to check whether X is a member of L or not.

**CODE:**

memb(X, [X|Tail]).

memb(X, [Head|Tail]):- memb(X, Tail).

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

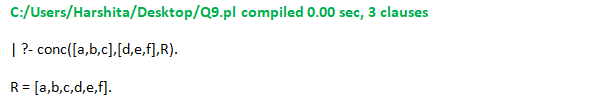
Q9. Write a Prolog program to implement conc (L1, L2, L3) where L2 is the list to be appended with L1 to get the resulted list L3.

**CODE**:

conc([],L,L).

conc([X|M],N,[X|Q]):- conc(M,N,Q).

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q10. Write a Prolog program to implement reverse (L, R) where List L is original and List R is reversed list.

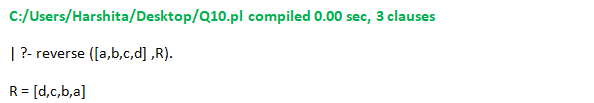
**CODE**:

reverse([H|T],R):- length(T,L),

L>0 ->(reverse(T,R1),R is H) ;

R is H.

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q11. Write a program in PROLOG to implement palindrome (L) which checks whether a list L is a palindrome or not.

**CODE**:

palind([]):- write('palindrome').

palind([\_]):- write('palindrome').

palind(L) :- append([H|T], [H], L),palind(T) ;

write('Not a palindrome').

**OUTPUT**:

****

Q12. Write a Prolog program to implement sumlist(L, S) so that S is the sum of a given list L.

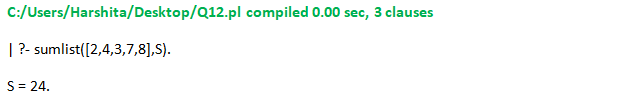
**CODE**:

sumlist([],0).

sumlist([H|T],S):- sumlist(T,S1),

S is H+S1.

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q13. 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.

**CODE**:

even\_length([]).

even\_length([\_|T]):- odd\_length(T).

odd\_length([\_]).

odd\_length([\_|T]):- even\_length(T).

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q14. 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.

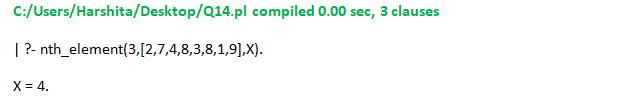
**CODE**:

nth\_element(1,[H|T],H).

nth\_element(N,[H|T],X):- N1 is N-1,

nth\_element(N1,T,X).

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q15. Write a Prolog program to implement maxlist(L, M) so that M is the maximum number in the list.

**CODE**:

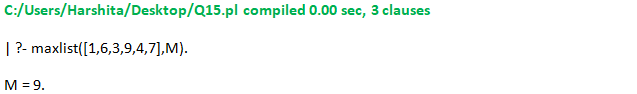
maxlist([H],H).

maxlist([H|T],M):- maxlist(T,M1),

H<M -> M is M1;

M is H.

**OUTPUT**:

****

**--------------------------------------------------------------------------------------------------------------------------------------**

Q16. 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.

**CODE**:

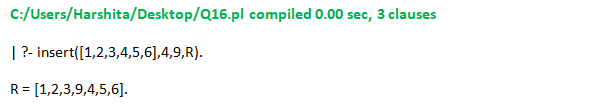
insert(L,1,Elem,[Elem|L]):-!.

insert([],\_,Elem,[Elem]).

insert([H|T],N,Elem,[H|R]):- C is N-1,

insert(T,C,Elem,R).

**OUTPUT**:



**--------------------------------------------------------------------------------------------------------------------------------------**

Q17. 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.

**CODE**:

delete(1,[\_|T],T).

delete(P,[X|Y],[X|R]):-

P1 is P-1, delete(P1,Y,R).

**OUTPUT**:



**--------------------------------------------------------------------------------------------------------------------------------------**