Practical File

Name – **Poonam**

Roll No. - **10858**

Course – B. Sc (Hons) Computer Science

Section – B

Semester - VI

Subject – Artificial Intelligence

INDEX

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.
	Extra Practicals
19.	Write a program in Prolog to implement delete the occurrence of the elements in the list.
20.	Write a program in Prolog to delete the element from the list .
21.	Write a program in Prolog to check the grammar of the given sentence. (NLP Program)

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

Code

```
sum :-write("Enter the first number "),
    read(X),
    write("Enter the second number "),
    read(Y),
    Z is X+Y,
    write(Z).
```

Output

```
% c:/Users/HP/Documents/Poonam AI/Poonam.pl compiled 0.02 sec, 51 clauses
?-
|    sum.
Enter the first number 53.
Enter the second number |: 2.
55
true.
```

Ques.2 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 ; M is Y.
```

Output

```
?- max(96,23,R).
R = 96.

?-
```

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

Code

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

Output

```
?- factorial(5,M).
M = 120 ,

?-
```

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

Code

```
generate_fib(1,0) :-!.
generate_fib(2,1).
generate_fib(N,T) :-N>0 ,N1 is N-1,N2 is N-2,generate_fib(N1,T1),generate_fib(N2,T2),T is T1+T2.
```

Output

```
?- generate_fib(5,R).
R = 3 ,
```

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

Code

```
gcd(X,Y,T) :- X=Y -> T is X.
gcd(X,Y,T) :- X>Y-> X1 is X-Y ,gcd(X1,Y,T1),T is T1; Y1 is Y-X ,gcd(X,Y1,T2),T is T2.
```

Output

```
?-
|
|    gcd(15,10,R).
R = 5 ;

   -
```

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

Code

```
power(0,0,0).
power(Num,0,1) :- Num>0.
power(Num,Pow,Ans) :- Num>0 ,Pow>0,P1 is Pow-1,power(Num,P1,R1),Ans is R1*Num.
```

Output

```
?- power(3,3,R).
R = 27 ;

^
```

Ques.7 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,R) :- R1 is N2-1,multi(N1,R1,R2),R is R2+N1.
```

Output

```
?- multi(8,5,R).
R = 40 ,
```

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

Code

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

Output

```
?-
|    memb(4,[5,8,9,6,7,2,4]).
true.

?- memb(14,[5,8,9,6,7,2]).
false.
```

Ques.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.

Code

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

Output

```
?- conc([3,4,5,2],[83,34,9,1],R).
R = [3, 4, 5, 2, 83, 34, 9, 1].
```

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

Code

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

Output

```
?- reverse([4,5,34,65,76],R).
R = [76, 65, 34, 5, 4].
```

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

Code

```
palindrome(L) :- reverse(L,L) ,L==L -> write("Given list is PALINDROME"); write("Given list is NOT PALINDROME").
```

Output

```
?- palindrome([1,3,1]).
Given list is PALINDROME
true.

?- palindrome([1,3,2]).
Given list is NOT PALINDROME
true.
```

Ques.12 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

```
?- sumlist([1,2,6,4,3],R).
R = 16.
```

Ques.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.

Code

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

Output

```
?- evenlength([3,4,5,6]).
true .

?- evenlength([3,4,5,6,43]).
false.

?- oddlength([3,4,5,6,43]).
true .

?- oddlength([3,4,5,43]).
false.
```

Ques.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.

Code

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

Output

```
?- nth_element(2,[45,34,23,54,65],X).
X = 34.
```

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

Code

```
maxm(X,Y,Z) :- X>Y ,Z is X; Z is Y.
maxlist([],0).
maxlist([R],R) :- !.
maxlist([H|T],M) :- maxlist(T,M1), maxm(H,M1,M).
```

Output

```
?- maxlist([34,54,65,35,69,99,74],R).
R = 99 ,
```

Ques.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.

Code

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

Output

```
?- insert_nth(6,3,[45,65,34,2,5,9,79],R).
R = [45, 65, 6, 34, 2, 5, 9, 79].
```

Ques.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.

Code

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

Output

```
?- delete_nth(2,[45,3,23,54,65,34],R).
R = [45, 23, 54, 65, 34].
```

Ques.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.

Code

```
merge(X, [], X).
merge([], Y, Y).
merge([X|X1], [Y|Y1], [X|Z]) :- X < Y, !, merge(X1, [Y|Y1], Z).
merge([X|X1], [Y|Y1], [X,Y|Z]) :- X == Y, !, merge(X1, Y1, Z).
merge([X|X1], [Y|Y1], [Y|Z]) :- X > Y, !, merge([X|X1], Y1, Z).
```

Output

```
?- merge([1,4,5,6],[8,26,45,50],R).
R = [1, 4, 5, 6, 8, 26, 45, 50].

?- merge([1,4,5,67],[34,65,76,97],R).
R = [1, 4, 5, 34, 65, 67, 76, 97].
```

Ques.19 Write a program in Prolog to implement delete the occurrence of the elements in the list.

Code

```
duplicates([],[]):- !.
duplicates([H|T],R):- memb(H,T), duplicates(T,R).
duplicates([H|T],[H|R]):- duplicates(T,R).
```

Output

```
?- duplicates([4,5,3,4,6,4],R).
R = [5, 3, 6, 4] ;

?- duplicates([4,5,3,4,6,4,5,4,6],R).
R = [3, 5, 4, 6]
```

Ques.20 Write a program in Prolog to delete the element from the list .

Code

```
delete_element(_, [], []).
delete_element(X, [X|T], T1) :-  delete_element(X, T, T1).
delete_element(X, [H|T], [H|T1]) :-delete_element(X, T, T1).
```

Output

```
?- delete_element(3,[4,5,3,7,8],R).
R = [4, 5, 7, 8] ;
```

Ques.21 Write a program in Prolog to check the grammar of the given sentence. (NLP Program)

Code

```
sentence(A,C):- nounPhrase(A, B), verbPhrase(B,C).

nounPhrase(A,C):- article(A,B), noun(B,C).
nounPhrase(A,B):- noun(A,B).
verbPhrase(A,C):- verb(A,B), prepositionPhrase(B,C).
verbPhrase(A,B):- verb(A,B).
verbPhrase(A,C):- verb(A,B), nounPhrase(B,C).

prepositionPhrase(A,C):- preposition(A,B), nounPhrase(B,C).

preposition([at|X],X).
article([a|X],X).
article([the|X],X).
noun([dog|X],X).
noun([cow|X],X).
noun([moon|X],X).
verb([barked|X],X).
verb([winked|X],X).
```

Output

```
?-
|    sentence([the,dog,barked,at,the,moon],R).
R = [] ;

?- sentence([barked,a,moon,dog,the],R).
false.

?-
```

**********End Of The Programs**********

Poonam

(10858)