Atma Ram Sanatan Dharma College

Artificial Intelligence Practical File

SUBMITTED BY

Name :vaivhav utreja

Course : Bsc.(Hons) Computer Science

Roll no : 22/28050

Semester : 6

Subject : Artificial Intelligence

Teacher : Dr. Parul Jain

Program 1

Write a PROLOG program to implement the family tree and demonstrate the family relationship.

~~CODE~~

Program

```
% Family tree representation
parent(john, mary).
parent(john, peter).
parent(mary, alice).
parent(peter, bob).
parent(robert, john).
parent(robert, linda).
parent(susan, mary).
parent(susan, peter).

male(john).
male(peter).
male(bob).
male(robert).

female(mary).
female(alice).
female(linda).
female(susan).

father(X, Y) :- parent(X, Y), male(X).
mother(X, Y) :- parent(X, Y), female(X).
sibling(X, Y) :- parent(Z, X), parent(Z, Y), X \= Y.

grandfather(X, Y) :- parent(X, Z), parent(Z, Y), male(X).
grandmother(X, Y) :- parent(X, Z), parent(Z, Y), female(X).
ancestor(X, Y) :- parent(X, Y).
ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y).

aunt(X, Y) :- sibling(X, Z), parent(Z, Y), female(X).
uncle(X, Y) :- sibling(X, Z), parent(Z, Y), male(X).
```

Output

grandfather(X, bob).

X = john

Next | 10 | 100 | 1,000 | Stop

grandmother(Y, alice).

Y = susan

Next | 10 | 100 | 1,000 | Stop

ancestor(A, bob).

A = peter

Next | 10 | 100 | 1,000 | Stop

Program 2

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~~

Program

```
% Q2 concatenation of lists
conc([],L,L).
conc([H|T],L,[H|R]):- conc(T,L,R).
```

Output

```
conc([1,2,3,4],[5,6],R).
R = [1, 2, 3, 4, 5, 6]
```

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

Program 3

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

~~CODE~~

Program

```
% Q3 Reversing a list
conc([],L,L).
conc([H|T],L,[H|R]):- conc(T,L,R).

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

Output

```
reverse([1,2,3,4],Rev).
Rev = [4, 3, 2, 1]
```

```
?- reverse([1,2,3,4],Rev).
```

Program 4

Write a PROLOG program to calculate the sum of two numbers.

~~CODE~~

Program

```
% Q4 Sum of two numbers
sum(A,B,C):- C is A+B.
```

Output

```
sum(4,5,C).
C = 9
```

```
?- sum(4,5,C).
```

Program 5

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

~~CODE~~

Program

```
% Maximum of two numbers
max(X, Y, X) :- X >= Y.
max(X, Y, Y) :- Y > X.
```

Output

```
max(5,10,S).
S = 10
```

```
?- max(5,10,S).
```

Program 6

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

~~CODE~~

Program

```
% Q6Factorial Calculation
factorial(0,1).
factorial(N,F):-
    N>0, N1 is N-1, factorial(N1,F1),
    F is N*F1.
```

Output

```
factorial(5,F).
F = 120
```

```
?- factorial(5,F).
```

Program 7

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

~~CODE~~

Program

```
% Q7 fibonacci number generation
fib(0,0).
fib(1,1).
fib(N,T):-
    N>1,N1 is N-1,N2 is N-2,
    fib(N1,T1),fib(N2,T2),T is T1+T2.
```

Output

```
fib(6,T).
T = 8
```

```
?- fib(6,T).
```

Program 8

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

~~CODE~~

Program

```
% Q8 Power function
power(_,0,1).
power(Num,Pow,Ans):-
  Pow > 0,Pow1 is Pow-1,
  power(Num,Pow1,Ans1),
  Ans is Num*Ans1.
```

Output

```
power(2,5,S).
S = 32
?- power(2,5,S).
```

Program 9

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

~~CODE~~

Program

```
% Q9 Multiplication of two numbers
multi(A,B,C):- C is A*B.
```

Output

```
multi(5,4,A).
A = 20
?- multi(5,4,A).
```

Program 10

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

~~CODE~~

Program

```
% Q10 check membership in a list
memb(X,[X|_]).
memb(X,[_|T]):- memb(X,T).
```

Output

```
memb(4,[1,2,3,5,5]).
false
memb(1,[1,2,3,5,5]).
true
```

```
?- memb(4,[1,2,3,5,5]).
memb(1,[1,2,3,4]).
```

Program 11

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

~~CODE~~

Program

```
% Q11 sum of a list
sumList([],0).
sumList([H|T],S):-
    sumList(T,S1), S is H+S1.
```

OUTPUT

```
sumList([1,2,3,4,5],A).
A = 15
```

```
?- sumList([1,2,3,4,5],A).
```

Program 12

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

Program

```
% Q12 even and odd length of the list
evenLength([]).
evenLength([_|[_|List]]):-
    evenLength(List).
oddLength([_]).
oddLength([_|[_|List]]):-
    oddLength(List).
```

OUTPUT

```
evenLength([1,2,3,4]).
true

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

```
evenLength([2,3,4]).
false

?- evenLength([2,3,4]).
```

```
oddLength([2,3,4]).
true

?- oddLength([2,3,4]).
```

Program 13

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

~~CODE~~

Program

```
% Q13 Maximum element in the List
maxList([M],M).
maxList([H|T],M):-
    maxList(T,M1),
    (H > M1 -> M=H; M=M1).
```

OUTPUT

```
maxList([4,5,10,11],S).
S = 11
```

```
?- maxList([4,5,10,11],S).
```

Program 14

Write a PROLOG program to implement insert(I, N, L, R) that inserts an item I into Nth position of list L to generate a list R.

~~CODE~~

Program

```
% Insert element at Nth position
insert(I, 1, L, [I|L]).
insert(I, N, [H|T], [H|R]) :-
    N > 1, N1 is N - 1,
    insert(I, N1, T, R).
```

OUTPUT

```
insert(2,3,[4,6,8,10],R).
R = [4, 6, 2, 8, 10]
```

```
?- insert(2,3,[4,6,8,10],R).
```

Program 15

Write a PROLOG program to implement delete(N, L, R) that removes the element on Nth position from a list L to generate a list R.

~~CODE~~

```
% Q15 Delete element at Nth Position
delete(1,[_|T],T).
delete(N,[H|T],[H|R]):-
    N > 1 , N1 is N-1, delete(N1,T,R).
```

OUTPUT

```
delete(2,[4,6,8,10],R).
R = [4, 8, 10]
```

```
?- delete(2,[4,6,8,10],R).
```