

LAB 4 | Artificial Intelligence

Aim : To learn arithmetic operations and recursion in Prolog.

1. Write a prolog program to find roots (real roots only) of quadratic equations.
-

Code:

```
predicates
    go.
    find_roots(real,real,real).

clauses
    go:-
        write("Enter a,b,c : "),nl,
        readreal(A),
        readreal(B),
        readreal(C),
        D = (B*B) - (4*A*C),
        find_roots(D,A,B).

    find_roots(D,A,B):-
        D=0,
        X=(-B)/(2*A),
        write("X = "),write(X),nl;

        D>0,
        X1=(-B + sqrt(D))/(2*A),
        X2=(-B - sqrt(D))/(2*A),
        write("x1 = "),write(X1),
        write(" x2 = "),write(X2),nl;

        D<0,
        write("Not possible to find roots."),nl.
```

Output:

Case I:

```
Goal : go
Enter a,b,c :
1
3
2
x1 = -1 x2 = -2
Yes
```

Case II:

```
Goal : go
Enter a,b,c:
1
0
1
Not possible to find roots.
Yes
```

- Write a prolog program to implement a logon routine. This routine must ask username and password and verify with a pair of username and password available (i.e. stored as clauses) as facts. On a successful match system display "welcome message" and on an unsuccessful attempt the user is allowed 3 times to re enter valid credentials. If a user enters incorrect credentials continuously 3 times then the system exits with "unsuccessful attempt message".

Code:

```
domains
    person, password = symbol

predicates
    login.
    count(integer).
    logon(person,password).
    message(integer).

clauses
    logon("Raj","raj5126").
    logon("Parth","parth5820").
    login:-
        write("Welcome ! Please Login"),nl,
        count(3),
        write("Login Unsuccessful"),nl.

    login:-write("Login Successful"),nl.

    count(X):-
        X<>0,
        write("Enter user name : "),nl,
        readln(User),
        write("Enter password : "),nl,
        readln(Passwd),
        not(logon(User,Passwd)),
        Y=X-1,
        message(Y),
        count(Y).
    count(0).

    message(X):-X<>0,write("Login Failed, Please Try Again Later"),nl.
    message(0).
```

Output:

Case I:

```
Goal : login
Welcome ! Please Login
Enter user name :
Raj
Enter password :
raj5126
Login Successful
Yes
```

Case II:

```
Goal : login
Welcome ! Please Login
Enter user name:
Dhruv
Enter password :
dp0105
Login Failed, Please Try Again Later
Enter user name:
Dhruv
Enter password :
dp2002
Login Failed, Please Try Again Later
Enter user name:
Parth
Enter password :
parth5820
Login Unsuccessful
Yes
```

3. Write a prolog program to find the factorial of a given number.
-

Code:

```

predicates
    factorial(integer,integer).
clauses
    factorial(0,1).
    factorial(N,F):-
        N>0,
        N1=N-1
        factorial(N1,F1),
        F=N*F1.

```

Output:

```

Goal : factorial(5,F)
F=120
1 Solution
Goal : factorial(0,F)
F=1
1 Solution

```

4. Write a prolog program to find the sum of first n numbers.
-

Code:

```

predicates
    sum(integer,integer).
clauses
    sum(0,0).
    sum(N,Sum):-
        N>0,
        N1=N-1,
        sum(N1,R1),
        Sum=R1+N.

```

Output:

```

Goal : sum(10,X)
X=55
1 Solution

```

5. Write a prolog program to print the nth term of Fibonacci series.
-

Code:

```
predicates
    fib(integer,integer).

clauses
    fib(0,0):-!.
    fib(1,1):-!.
    fib(N,Result):-
        N1=N-1,
        N2=N-2,
        fib(N1,Result1),
        fib(N2,Result2),
        Result=Result1+Result2.
```

Output:

```
Goal : fib(2,X)
X=1
1 Solution
Goal : fib(7,X)
X=13
1 Solution
```

6. Write a prolog program to print Fibonacci series up-to nth term.

Code:

```
predicates
    fib(integer,integer,integer).
go.

clauses
    go:-
        write("Enter number : "),
        readreal(N),
        A=0,
        B=1,
        write(A),write(' '),write(B),write(' '),
        fib(N,A,B).

    fib(N,A,B):-
        N<2,nl;
        C=A+B,
        write(C),
        write(" "),
        D=B,
        E=C,
        N1=N-1,
        fib(N1,D,E).
```

Output:

```
Goal : go
Enter number : 5
0 1 1 2 3 5
Yes
Goal : go
Enter number : 1
0 1
Yes
```