

ARTIFICIAL INTELLIGENCE

LAB 4

AIM: To learn arithmetic operations and recursion in prolog.

EXERCISE:

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

Code:

```
predicates
    go
    roots(real,real,real)
clauses
go:-
    write("Enter the value of a,b,c : "),nl,
    readreal(A),
    readreal(B),
    readreal(C),
    D = (B*B)-(4*A*C),
    roots(D,A,B).
roots(D,A,B):-
    D=0,
    ROOT = (-B) / (2*A),
    write("Unique Root = "),
    write(ROOT),nl;
    D>0,
    SQRT1=sqrt(D),
    R1 = (-B + SQRT1)/(2*A),
    R2 = (-B - SQRT1)/(2*A),
    write("Root 1 = ",R1),nl,
    write("Root 2 = ",R2),nl;
    D<0,
    write("No real Roots").|
```

Output:

Goal: go
Enter the value of a, b, c:
1
11
-12
Root 1 = 1
Root 2 = -12
Yes

Goal: go
Enter the value of a, b, c:
1
7
6
Root 1 = -1
Root 2 = -6
Yes

Goal: go
Enter the value of a, b, c:
1
6
30
No real Roots
Yes

2. 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 reenter valid credentials. If a user enters incorrect credentials continuously 3 times then the system exits with “unsuccessful attempt message”.

Code:

```
domains
    username , password = symbol
predicates
    person(username,password)
    login
    count(integer)
    msg(integer)
clauses
    person(abc,abc_123).
    person(xyz,xyz_123).
    login :-
        write("Welcome! Please Login"),nl,
        count(3),
        write("Login Unsuccessful"),nl.
    login :-
        write("Login Successful"),nl.
    count(X) :-
        X <> 0,
        write("Enter Username :- "),
        readln(Username),
        write("Enter Password :- "),
        readln>Password),
        not(person(Username>Password)),
        Y = X - 1,
        msg(Y),
        count(Y).
    count(0).
    msg(X) :-
        X <> 0,
        write("Login Failed\nPlease Try Again!\n").
    msg(0).
```

Output:

1. Login Unsuccessful

```
Goal: login
Welcome! Please Login
Enter Username :- abc
Enter Password :- abc

Login Failed!

Please Try Again!
Enter Username :- xyz
Enter Password :- abc_123

Login Failed!

Please Try Again!
Enter Username :- abc
Enter Password :- abc_12

Login Unsuccessful
Yes
```

2. Login successful

```
Goal: login
Welcome! Please Login
Enter Username :- abc
Enter Password :- abc

Login Failed!

Please Try Again!
Enter Username :- abc
Enter Password :- abc_123

Login Successful
Yes
```

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

Code:

```
predicates
    go
    factorial(integer,integer)
clauses
    go:-
        write("Enter the positive number = "),
        readreal(Num),
        Ans = 1.0,
        factorial(Num,Ans).
    factorial(Num,Ans):-
        Num <> 0,
        NewResult = Num * Ans,
        NN = Num - 1,
        factorial(NN,NewResult).
    factorial(_,Ans):-
        write("Factorial = ",Ans),nl.
```

Output:

```
Goal : go
Enter the positive number = 6
Factorial = 720
Yes

Goal : go
Enter the positive number = 3
Factorial = 6
Yes
```

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

Code:

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

Output:

Goal: sum(3, Sum)
Sum=6

Goal: sum(10, Sum)
Sum=55

Goal: sum(5, Sum)
Sum=15

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

Code:

```
predicates
    fibo(integer,integer)
    fibo(integer)
clauses
    fibo(1,1).
    fibo(2,1).
    fibo(N,F):-
        N>2,
        N1=N-1,
        fibo(N1,F1),
        N2=N-2,
        fibo(N2,F2),
        F=F1+F2.
    fibo(N):-
        fibo(N,F),
        write(F),nl.
```

Output:

```
Goal : fibo(4)
3
Yes
Goal : fibo(3)
2
Yes

Goal : fibo(10)
55
Yes
```

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

Code:

```
domains
    C,N,N1,E,A,B,D,E=Integer
predicates
    fibo(integer,integer,integer)
go
clauses
    go:-
        write("Enter a number : "),readint(N),nl,
        write("Fibonacci series for "),write(N),write(" elements is : "),nl,
        A = 0,
        B = 1,
        write(A),write(" "),write(B),write(" "),
        fibo(N,A,B).
    fibo(N,A,B):-
        N<2,nl;
        C = A+B,
        write(C),write(" "),
        D = B,
        E = C,
        N1 = N-1,
        fibo(N1,D,E).
```

Output:

```
Goal : go
Enter a number : 4
Fibonacci series for 4 elements is :
0 | 1 | 2
Yes

Goal : go
Enter a number : 6
Fibonacci series for 6 elements is :
0 | 1 | 2 | 3 | 5
Yes

Goal : go
Enter a number : 1
Fibonacci series for 1 elements is :
0
Yes
```