ARTIFICIAL INTELLIGENCE

LAB 5

AIM: To study about controlling execution in prolog using cut and fail predicate.

EXERCISE:

1. Implement a prolog program to find minimum and maximum of two integers using cut and/or fail predicate. Program must have three arguments and it must handle all cases.

Code:

```
predicates
    max(integer,integer,integer)
    min(integer,integer,integer)
clauses
    max(X,Y,Z):-
        X>=Y,
        Z=X,
        !.
    max(X,Y,Z):-
        X<Y,
        Z=Y.
    min(X,Y,Z):-
        X<=Y,
        Z=X,
        !.
    min(X,Y,Z):-
        X>Y,
        Z=Y.
```

Output:

```
Goal : max(5,10,T)
T=10
1 Solution

max(10,20,20)
Yes

max(5,10,5)
No

min(7,4,T)
T=4
1 Solution

min(1,2,1)
Yes
```

2. Write a prolog program to verify that a given year is leap year or not using cut and/or fail predicate.

Note: A year is a leap year if it is divisible by 4, but century years are not leap years unless they are divisible by 400. So, the years 1700, 1800, and 1900 were not leap years, but the year 2000 was.

Code:

```
domains
    z=integer
predicates
    leap_year(integer)
clauses
    leap_year(X):-
        X mod 4<>0,
        !,
        fail.
    leap_year(X):-
        X mod 100<>0,
        !.
    leap_year(X):-
        X mod 400=0.
```

Output:

```
Goal : leap_year(8000)
Yes

Goal : leap_year(2026)
No

Goal : leap_year(2020)
Yes

Goal : leap_year(1700)
No
```

3. Write a prolog program to verify whether a given number is prime or not using cut and/or fail predicate.

Code:

```
predicates
    check(integer,integer)
    isPrime(integer)
clauses
    check(X,Y) :-
        X mod Y = 0,
        !.
    check(X,Y) :-
        X > Y+1,
        Y=Y+1,
        check(X, Y).
    isPrime(2) :-
        !.
    isPrime(X) :-
        X < 2,
        !,
        fail.
    isPrime(X) :-
        not(check(X, 2)).
```

Output:

```
Goal : isPrime(5)
Yes

Goal : isPrime(-1)
No

Goal : isPrime(10)
No
```