Logic Programming with Prolog

Section 4

Created by: Rehab Mohamed

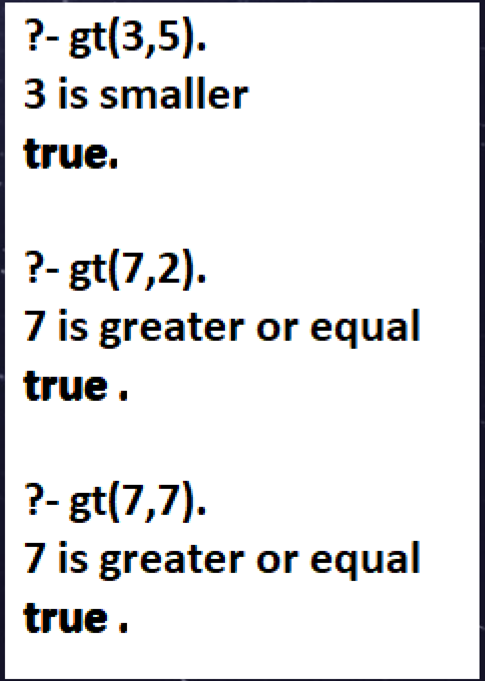
Decision Making

The decision statements are If-Then-Else statements. So, when we try to match some condition, and perform some task, then we use the decision-making statements.

Example

```
% If-Then-Else statement

gt(X,Y) :- X >= Y,write(X),write(' is greater or equal').
gt(X,Y) :- X < Y,write(X),write(' is smaller').
```



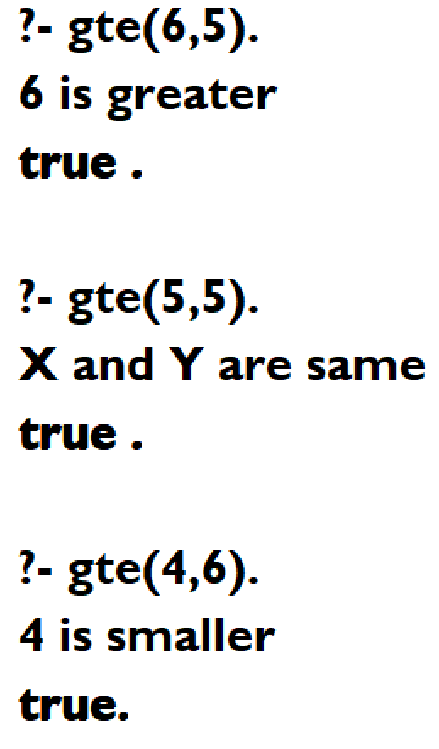
```
?- gt(3,5).
3 is smaller
true.

?- gt(7,2).
7 is greater or equal
true .

?- gt(7,7).
7 is greater or equal
true .
```

```
% If-Else statement

gte(X,Y) :- X > Y,write(X),write(' is greater').
gte(X,Y) :- X =:= Y,write('X and Y are same').
gte(X,Y) :- X < Y,write(X),write(' is smaller').
```



```
?- gte(6,5).
6 is greater
true .

?- gte(5,5).
X and Y are same
true .

?- gte(4,6).
4 is smaller
true.
```

Loops

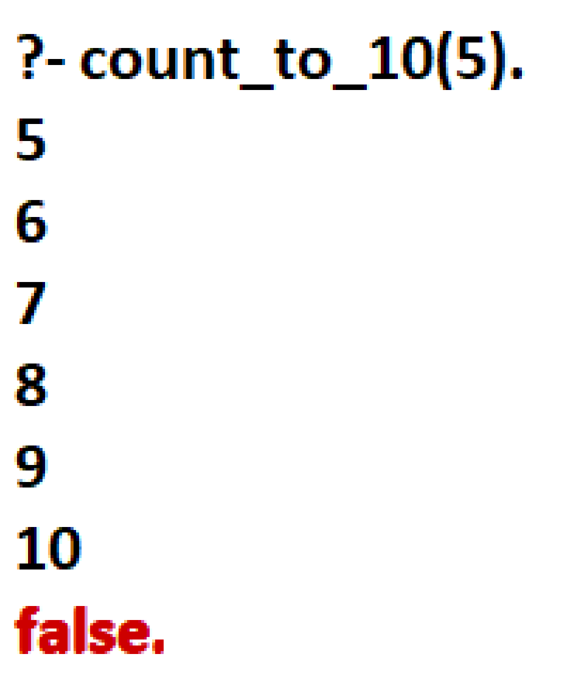
Many programming languages provide 'for loops' which enable a set of instructions to be executed a fixed number of times. No such facility is available in Prolog (directly), but a similar effect can be obtained using recursion. Loop statements are used to execute the code block multiple times.

Example 1: Output integers from a specified value down to 1.

```
loop(0):- writeln(“End of example”).
loop(N):- N>0, writeln(N), M is N-1,loop(M).
```

Example 2: Count from any number to 10

```
count_to_10(N):- N=<10,writeln(N),M is N+1,count_to_10(M).
```



```
?- count_to_10(5).
5
6
7
8
9
10
false.
```

Recursion

Prolog doesn't have iteration, but all iteration can be rewritten using recursion. Recursion is a function that can call itself until the goal succeeds. In Prolog, recursion appears when a predicate contain a goal that refers to itself.

Syntax:

```
rec(0) :- !.
rec(I) :- I1 is I – 1, rec(I1).
```

Example

The best way in Prolog to calculate a factorial is to do it recursively. Here is an example of how it can be done :

```
factorial(0,1):- !.
factorial(X,Result) :- X > 0, X1 is X - 1,
                      factorial(X1,Z),
                      Result is Z*X.
```

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

Task

1. Write a program in prolog to check number is Even or Odd.
2. Write a program in prolog to print odd numbers between 1 to N
3. Find the sum of the integers from 1 to N