

Principle of Programming Language

Assignment - 6

Adm. No.: U19CS082

Name: SOURABH PATEL

1. Write a program in Prolog that uses the following predicates Write, nl, read, consult, halt, statistics.

```
cube :-
    write ( 'Write a number: ' ),
    read (Number),
    process(Number).
process(stop) :- !.
process(Number) :-
    C is Number * Number * Number,
    write ( 'Cube of ' ), write (Number), write ( ': ' ), write (C),nl, cube.

?- consult('Q1.pl').
true.

?- cube.
Write a number: 1.
Cube of 1: 1
Write a number: |: 2.
Cube of 2: 8
Write a number: |: 3.
Cube of 3: 27
Write a number: |: stop.

true.

?- statistics.
% Started at Mon Feb 21 14:30:18 2022
% 0.089 seconds cpu time for 275,386 inferences
% 5,723 atoms, 4,292 functors, 3,117 predicates, 40 modules, 122,982 VM-codes
%
%
% Local stack:      Limit   Allocated   In use
% Global stack:     -       140 Kb      64 Kb
% Trail stack:      -       46 Kb       440 b
% Total:            1,024 Mb  206 Kb      66 Kb
%
% 2 garbage collections gained 186,288 bytes in 0.000 seconds.
% 2 atom garbage collections gained 657 atoms in 0.001 seconds.
% 5 clause garbage collections gained 123 clauses in 0.000 seconds.
% Stack shifts: 2 local, 4 global, 3 trail in 0.000 seconds
% 2 threads, 0 finished threads used 0.000 seconds
true.

?- halt.
```

2. Try to answer the following questions first “by hand” and then verify your answers using a Prolog interpreter.

- (a) Which of the following are valid Prolog atoms? `f`, `loves(john, mary)`, `Mary`, `_c1`, `'Hello'`, `this_is_it`

```
?- atom(f).
true.

?- atom(loves(john,mary)).
false.

?- atom(Mary).
false.

?- atom(_c1).
false.

?- atom('Hello').
true.

?- atom(this_is_it).
true.
```

- (b) Which of the following are valid names for Prolog variables? `a`, `A`, `Paul`, `'Hello'`, `a_123`, `_`, `_abc`, `x2`

A variable is a string of upper-case letters, lower-case letters, digits, and underscore characters that start either with an upper-case letter or with an underscore.

`A`, `Paul`, `_`, `_abc` are valid variable names.

- (c) What would a Prolog interpreter reply to given the following query? `?- f(a, b) = f(X, Y).`

```
?- f(a,b) = f(X,Y).
X = a,
Y = b.
```

- (d) Would the following query succeed?
`?- loves(mary, john) = loves(John, Mary).` Why?

Yes, the query will be successfully executed as it takes `mary` and `john` as arguments while `John` and `Mary` of second functions will be variable so it becomes the case of `Q(C)` which is given above.

```
?- loves(mary, john) = loves(John, Mary).
John = mary,
Mary = john.
```

- (e) Assume a program consisting only of the fact `a(B, B).` has been consulted by Prolog. How will the system react to the following query?

?- a(1, X), a(X, Y), a(Y, Z), a(Z, 100). Why?

The given function a(X, Y) is explained as X=Y.

here, the definition of function a(X, Y) is a(B, B) implies both the parameter are equal which gives the following output.

a(1,X) => X = 1. a(X,Y) => Y = X => Y =
1. a(Y,Z) => Z = Y => Z = 1. a(Z,100) =>
Z=100 => 1=100 return false.

```
?- a(1, X), a(X, Y), a(Y, Z), a(Z, 100).  
false.
```

3. Read the section on matching again and try to understand what's happening when you submit the following queries to Prolog.

- (a) ?- myFunctor(1, 2) = X, X = myFunctor(Y, Y).
- (b) ?- f(a, _, c, d) = f(a, X, Y, _).
- (c) ?- write('One '), X = write('Two ').

```
?- myFunctor(1, 2) = X, X = myFunctor(Y, Y).  
false.  
  
?- f(a, _, c, d) = f(a, X, Y, _).  
false.  
  
?- write('One '), X = write('Two ').  
One  
X = write('Two ').
```

4. Draw the family tree corresponding to the following Prolog program.

```
%female  
female(mary).  
female(sandra).  
female(juliet).
```

```
female(lisa).
```

```
%male
male(peter).
male(paul).
male(dick).
male(bob).
male(harry).

%parent
parent(bob, lisa).
parent(bob, paul).
parent(bob, mary).
parent(juliet, lisa).
parent(juliet, paul).
parent(juliet, mary).
parent(peter, harry).
parent(lisa, harry).
parent(mary, dick).
parent(mary, sandra).

%father
father(X,Y) :-
    male(X),
    parent(X,Y),!.

%sister
sister(X,Y) :-
    female(X),
    parent(Z,X),
    parent(Z,Y),
    X \= Y.

%brother
brother(X,Y) :-
    male(X),
    parent(Z,X),
    parent(Z,Y),
```

```
X \= Y.
```

```
%grandmother
```

```
grandmother(X,Y) :-
```

```
female(X),
```

```
parent(X,Z),
```

```
parent(Z,Y),!.
```

```
%cousin
```

```
cousin(X,Y) :-
```

```
parent(Z,X),
```

```
parent(W,Y),
```

```
(brother(Z,W);sister(Z,W)).
```

```
?- consult('Q4.pl').  
true.
```

```
?- father(bob,paul).  
true.
```

```
?- father(X,paul).  
X = bob.
```

```
?- sister(sandra,dick).  
true.
```

```
?- sister(dick,sandra).  
false.
```

```
?- grandmother(X,sandra).  
X = juliet.
```

```
?- grandmother(X,dick).  
X = juliet.
```

```
?- cousin(harry,dick).  
true .
```
