**Name : Brijesh Rameshbhai Rohit**

**Admission number: U19CS009**

# **PINCIPLES OF PROGRAMMING LANGUAGES**

## **Assignment - 06**

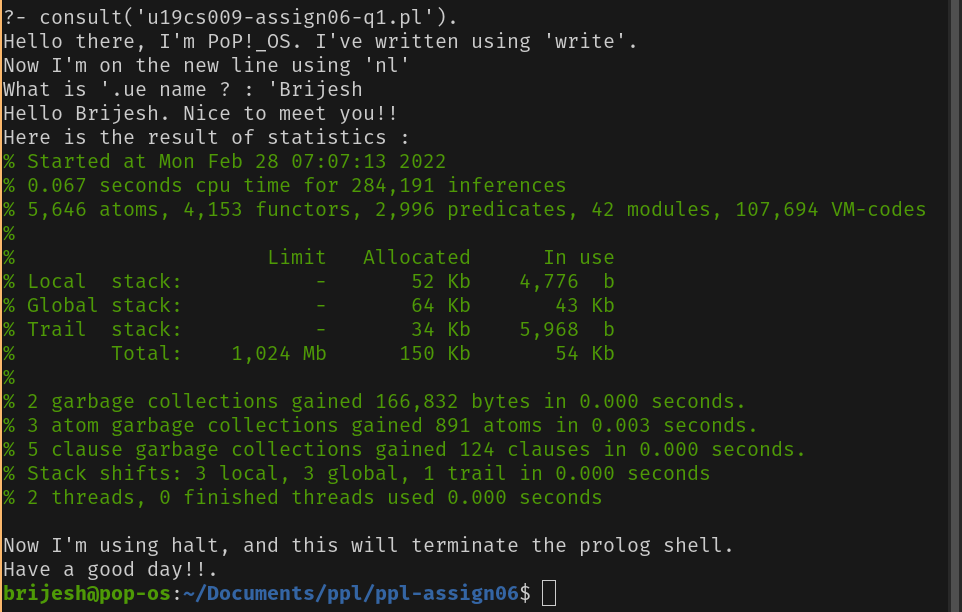
1.Write a program in Prolog that uses following predicates

Write, nl, read, consult,halt,statistics.

CODE=>

|  |
| --- |
| %U19CS009  %BRIJESH ROHIT  %start when program is consulted  :-  write("Hello there, I'm PoP!\_OS. I've written using 'write'."),  nl, write("Now I'm on the new line using 'nl'"),  write("\nWhat is youe name ? : "), read(Name),  write("Hello "),write(Name),write(". Nice to meet you!!"),nl,  write("Here is the result of statistics : "),  statistics,nl,  write("Now I'm using halt, and this will terminate the prolog shell.\nHave a good day!!."),nl,  halt. |

OUTPUT=>



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

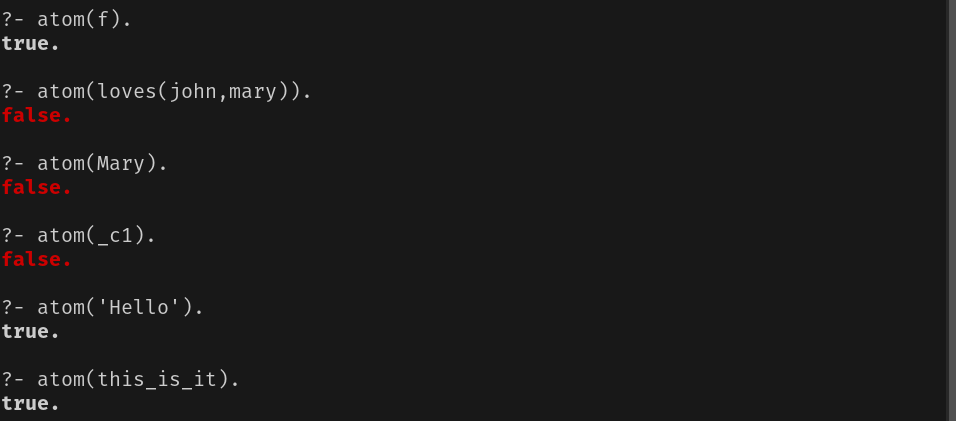
**(a) Which of the following are valid Prolog atoms?**

**f, loves(john,mary), Mary, \_c1, 'Hello', this\_is\_it**

**Ans: atoms are : f, ‘Hello’, this\_is\_it,**

**loves(john,mary) is fact,**

**Mary, \_c1 are variables.**



**(b) Which of the following are valid names for Prolog variables?**

**a, A, Paul, 'Hello', a\_123, \_, \_abc, x2**

**Ans: atoms are : A, Paul, \_,\_abc,**



**(c) What would a Prolog interpreter reply given the following query?**

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

**Ans: X=a,**

**Y=b.**

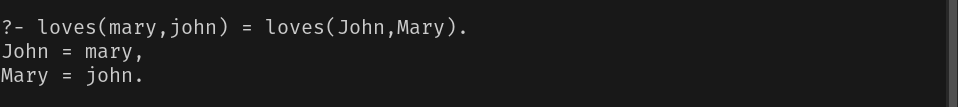


**(d) Would the following query succeed?**

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

**Ans: John=mary,**

**Mary=john.**



**Why?**

**Ans: This is because initially John and Mary are variables which do not**

**hold any value. Thus, the first ‘=’ would act as assignment rather**

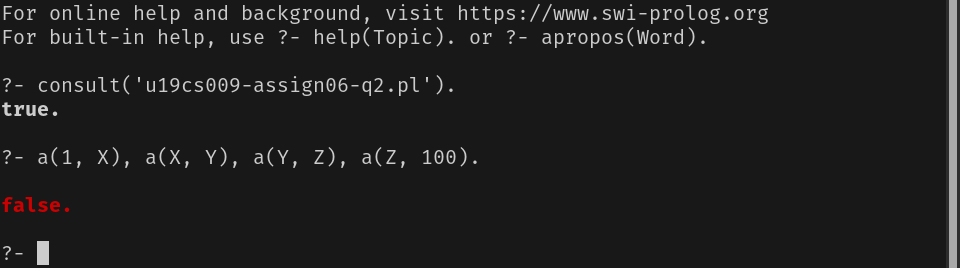
**than comparison and John would store mary, Mary will store john.**

**(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).**

**Ans: False.**



**Why?**

**Ans: As a(B,B). is the only fact in the knowledge base, it will return true**

**only when Both the variables store same value.**

**When we use a(1, X), a(X, Y), a(Y, Z), a(Z, 100). as command,**

**Because of 1st call X would hold value 1,**

**And Y will also hold value 1 because of call 2,**

**And Z will also hold value 1 because of call 3,**

**Now since Z is not empty variable anymore, a(Z,100) would now**

**perform comparison rather than assignment thus it would check if**

**a(Z,100) exist in the knowledge base or not.**

**a(Z,100) is now same as a(1,100) which would never exist in the**

**knowledge base due a(B,B) being the only fact.**

**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).**

**Ans: X would hold myFunctor(1,2) as value but when we write X= myFuntor(Y,Y) next instead of assignment comparison happens and because the two parameters inside myFunctor should be different and not the same (Y,Y) for X we get false as result.**

**If we replace the query X=myFunctor(Y,Y) with X=myFunctor(Y,Z)**

**then Y would hold value 1 and Z would hold value 2. Hence the result would be :**

**Y=1,**

**Z=2.**

**(b) ?- f(a, \_, c, d) = f(a, X, Y, \_).**

**Ans: The result would be Y=c. Since comparison is taking place, each**

**parameter will be compared.**

**1. 1st parameter is same in both the predicates.**

**2. 2nd parameter of LHS is '\_'meaning any value thus does not matter what parameter is there in RHS.**

**3. 3rd parameter in LHS is 'c' which would be assigned to variable 'Y'.**

**4. 4th parameter of RHS is '\_'again ignored.**

**Hence output would be:**

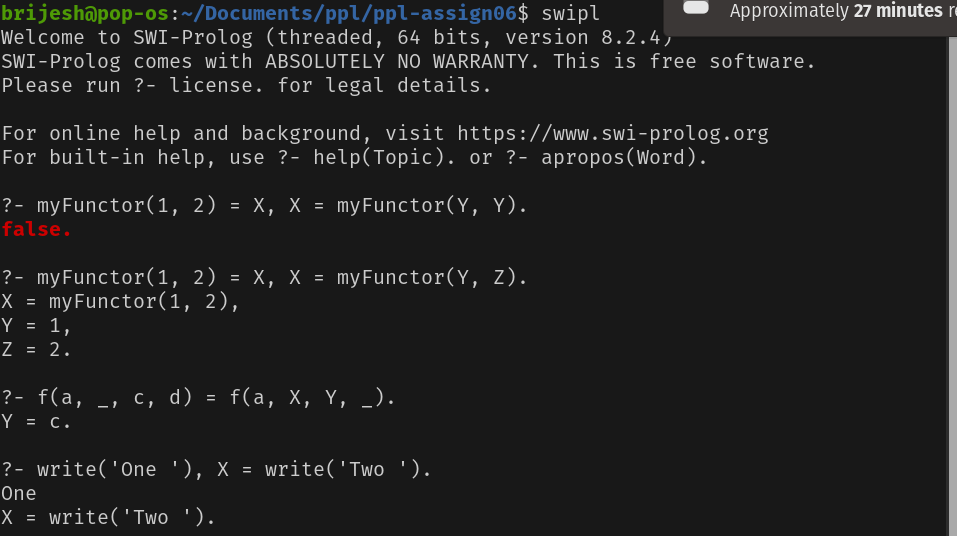
**Y = c.**

**(c) ?- write('One '), X = write('Two ').**

**Ans: The write('two') would be assigned to X instead of being executed. Hence the result would be :**

**One**

**X=write('Two').**



**4. Draw the family tree corresponding to the following Prolog program:female(mary).**

**female(sandra).**

**female(juliet).**

**female(lisa).**

**male(peter).**

**male(paul).**

**male(dick).**

**male(bob).**

**male(harry).**

**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).**

**After having copied the given program, define new predicates (in terms of rules using male/1,  female/1 and parent/2) for the following family relations:**

**(a) father**

**(b) sister**

**(c) grandmother**

**(d) cousin**

**You may want to use the operator \=, which is the opposite of =. A goal like X \= Y succeeds, if  the two terms X and Y cannot be matched.**

**Example: X is the brother of Y, if they have a parent Z in common and if X is male and if X and  Y don't represent the same person. In Prolog this can be expressed through the following rule:**

**brother(X, Y) :-**

**parent(Z, X),**

**parent(Z, Y),**

**male(X),**

**X \= Y.**

**CODE=>**

|  |
| --- |
| female(sandra).  female(juliet).  female(lisa).  male(peter).  male(paul).  male(dick).  male(bob).  male(harry).  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).  %New Predicates  %father predicate (X is father of Y)  father(X,Y):-  male(X),  parent(X,Y),  write(X), write(" is Father of "),write(Y),nl.  %sister predicate (X is sister of Y)  sister(X,Y):-  female(X),  parent(Z,X),  parent(Z,Y),  X\==Y,  write(X),write(" is Sister of "),write(Y),nl.  %grandmother predicate (X is grandmother of Y)  grandmother(X,Y):-  female(X),  parent(X,Z),  parent(Z,Y),  write(X), write(" is Grandmother of "),write(Y),nl.  %brother predicate (X is brother of Y)  brother(X,Y):-  male(X),  parent(Z,X),  parent(Z,Y),  X\==Y,  write(X), write(" is Brother of "),write(Y),nl.  %sibling predicate (X is sibling of Y)  sibling(X,Y):-  sister(X,Y);  brother(X,Y),  write(X), write(" is Sibling of "),write(Y),nl.  %cousin predicate (X is cousin of Y)  cousin(X,Y):-  parent(Z,X),  parent(W,Y),  sibling(Z,W),  write(X), write(" is Cousin of "),write(Y),nl. |

