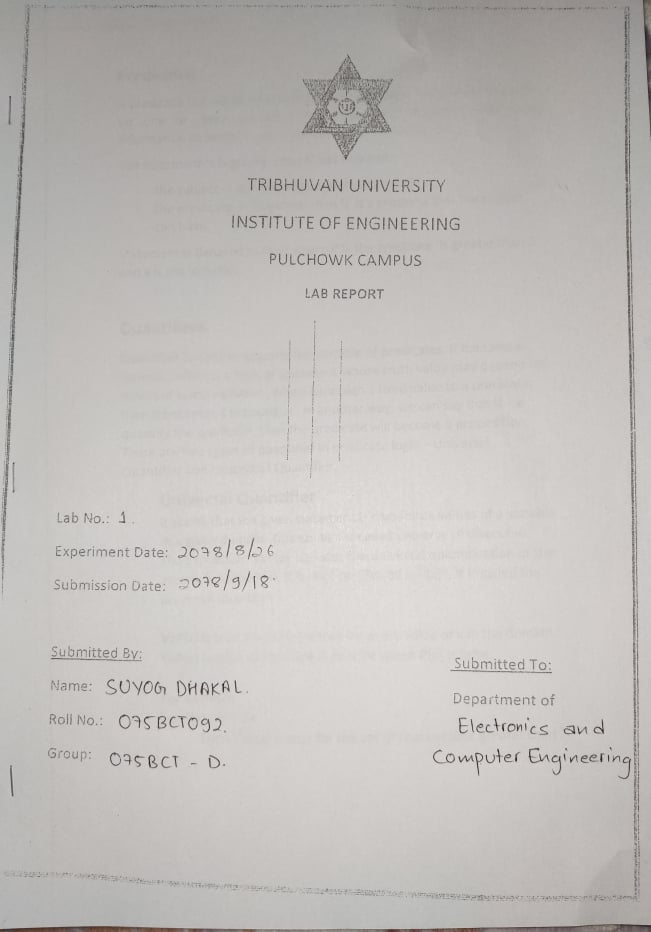
****

**Prolog**

Prolog is a very important tool in programming artificial intelligence applications and in the development of expert systems. Prolog as the name itself suggests, is the short form of LOGical PROgramming. It is a logical and declarative programming language. Prolog can be viewed as a tool to solve problems in the field of artificial intelligence or it can be very well used a general programming language.

**Data types in prolog**

• Atoms and numbers

• Variables

• Structures

**Atoms and numbers**

Atoms can be constructed in three different ways

1. strings of letters, digits, and the underscore character ‘\_’ starting with a lower case letter.

for example:

man, ram, comp\_students, pc\_ct\_059.

2. strings of special characters

for example:

<------->

:::::::::::

Care should be taken not to use the character combination that may have some built in meaning.

3. strings of characters enclosed in quotes

for example

‘Ram’

‘Bird’

Numbers used in prolog are integers and real numbers.

**Variables**

Variables are strings of letters digits and underscore that start with an underscore or an upper-case letter. The scope of a variable is one clause only. So the same variable used in different clauses mean different thing.

For example:

X, Ram, \_weight etc.

Note here that Ram is a variable unlike the earlier use ‘Ram’ where it was a constant, an atom.

An underscore ‘\_’ also known as anonymous variable is used in clauses when a variable need not be inferred to more than once.

**Structures**

Structures are objects that have different components. The components can be atoms or yet some other structures. A functor is used to construct a structure as follows.

family(father, mother, children)

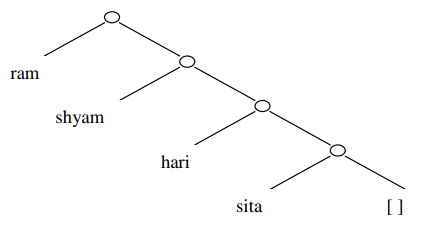
Here family is a structure that has father, mother and the children as its elements. The father and mother may be atoms while the children may be yet another structure or a list of atoms. List is a special built in structure in prolog.

A list is a built in structure in prolog. It can be thought of as a sequence of elements ordered linearly however it is internally represented as a binary tree.

For example:

[ram,shyam,hari,sita]

The representation is as follows



The list as such can be broken down into two parts, the HEAD and the TAIL. The head is the first element of the list and the tail is the remaining list. The above list can be broken down as

[H| T]

Where H= ram and T= [shyam, hari,sita]

List is one of the most useful structure in prolog.

**Example of program in Prologs**

% program that finds or checks the larger number

PREDICATES   
bigger(integer,integer,integer)

CLAUSES   
bigger(X,Y,Z):-   
 X>Y,Z=X.   
bigger(X,Y,Z):-   
 X<Y,Z=Y.

GOAL   
bigger(10,70,X).

## Predicate Logic – Definition

A predicate is an expression of one or more variables defined on some specific domain. A predicate with variables can be made a proposition by either assigning a value to the variable or by quantifying the variable.

The following are some examples of predicates −

* Let E(x, y) denote "x = y"
* Let X(a, b, c) denote "a + b + c = 0"
* Let M(x, y) denote "x is married to y"

## Well Formed Formula

Well Formed Formula (wff) is a predicate holding any of the following −

1.All propositional constants and propositional variables are wffs.

2.If x is a variable and Y is a wff, ∀xY∀xY and ∃xY∃xY are also wff.

3.Truth value and false values are wffs.

4.Each atomic formula is a wff.

5.All connectives connecting wffs are wffs.

## Quantifiers

The variable of predicates is quantified by quantifiers. There are two types of quantifier in predicate logic − Universal Quantifier and Existential Quantifier.

### Universal Quantifier

Universal quantifier states that the statements within its scope are true for every value of the specific variable. It is denoted by the symbol ∀∀.

∀xP(x)∀xP(x) is read as for every value of x, P(x) is true.

**Example** − "Man is mortal" can be transformed into the propositional form ∀xP(x)∀xP(x) where P(x) is the predicate which denotes x is mortal and the universe of discourse is all men.

### Existential Quantifier

Existential quantifier states that the statements within its scope are true for some values of the specific variable. It is denoted by the symbol ∃∃.

∃xP(x)∃xP(x) is read as for some values of x, P(x) is true.

**Example** − "Some people are dishonest" can be transformed into the propositional form ∃xP(x)∃xP(x) where P(x) is the predicate which denotes x is dishonest and the universe of discourse is some people.

### Nested Quantifiers

If we use a quantifier that appears within the scope of another quantifier, it is called nested quantifier.

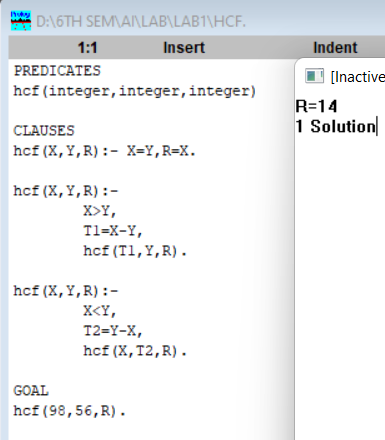
**Example**

1. ∀ a∃bP(x,y)∀ a∃bP(x,y) where P(a,b)P(a,b) denotes a+b=0a+b=0.
2. ∀ a∀b∀cP(a,b,c)∀ a∀b∀cP(a,b,c) where P(a,b)P(a,b) denotes a+(b+c)=(a+b)+ca+(b+c)=(a+b)+c.

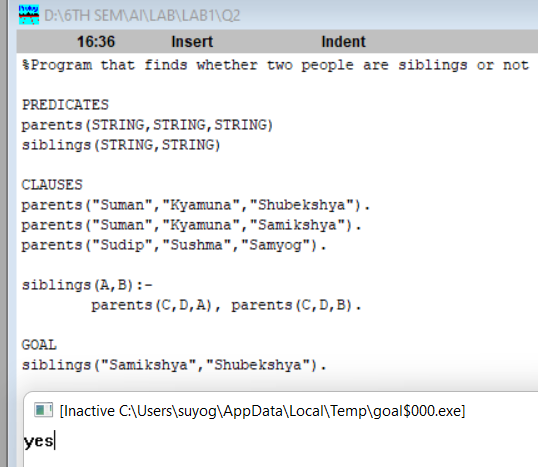
**Note** − ∀a∃bP(x,y)≠∃a∀bP(x,y)

Programs:

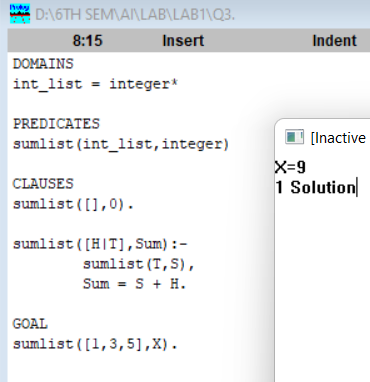
1. Write a program to find the HCF of two numbers.



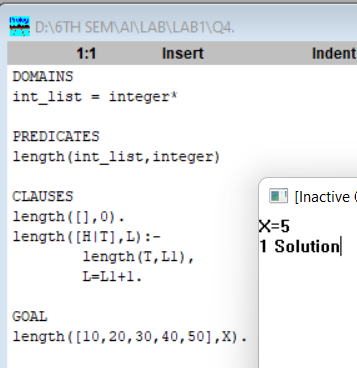
2.



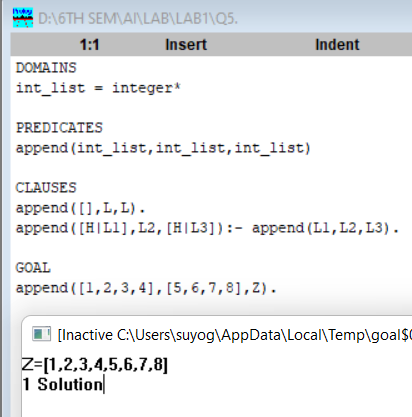
3.Write a program to add content of an integer list and display it.



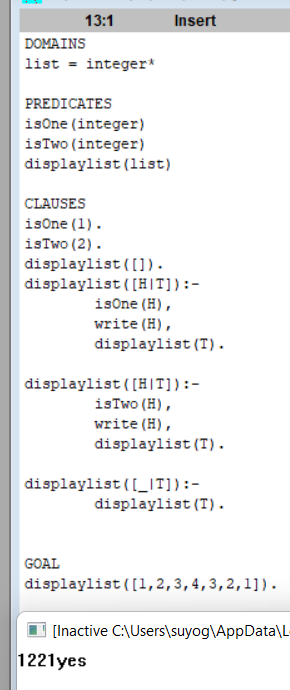
1. Write a program to find length of a list.



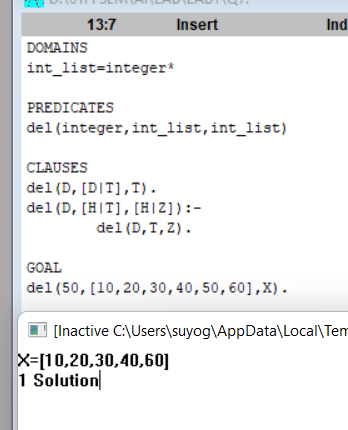
1. Write a program to append two list.



1. Write a program which takes lists of integer and display 1s and 2s only.



1. Write a program to delete an element from a list.



Discussion

In this lab we learned about the basics of Prolog. We learned about domains, predicates, clauses and many more syntax. After reading the content provided on labsheet and visualizing the given examples on visual prolog, we solved the assignment.

Conclusion

Hence, we learned about prolog and wrote the program of prolog on visual prolog software.