TRIVUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

Artificial Intelligence

LAB-1

Submitted to:
Dr. Basanta Joshi
Department of Electronics
And Computer Engineering

Submitted By:
Prakash Chaulagain
076BCT045

Date of Submission:
13th January 2023

Introduction to Prolog

Prolog or PROgramming in LOGics is a logical and declarative programming language. It is one major example of the fourth generation language that supports the declarative programming paradigm. This is particularly suitable for programs that involve symbolic or non-numeric computation. This is the main reason to use Prolog as the programming language in Artificial Intelligence, where symbol manipulation and inference manipulation are the fundamental tasks.

In Prolog, we need not mention the way how one problem can be solved, we just need to mention what the problem is, so that Prolog automatically solves it. However, in Prolog we are supposed to give clues as the solution method.

Prolog language basically has three different elements −

Facts − The fact is predicate that is true, for example, if we say, “Tom is the son of Jack”, then this is a fact.

Rules − Rules are extinctions of facts that contain conditional clauses. To satisfy a rule these conditions should be met.

Data types in prolog:

Atoms and numbers

Atoms can be constructed in three different ways

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

for example: man, ram, comp_students, pc_ct_059.

- strings of special characters for

example:

<-------> ::::::::::

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

- 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 means different things.

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 builtin 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]

Key Features :

1. Unification : The basic idea is, can the given terms be made to representthe same structure.
2. Backtracking : When a task fails, prolog traces backwards and tries tosatisfy the previous task.
3. Recursion : Recursion is the basis for any search in a program.

Running queries :

A typical prolog query can be asked as :

Query 1 : ?-
singer(sonu). Output :
Yes.

Explanation : As our knowledge base contains the above fact, so output was 'Yes', otherwise it would have been 'No'.

Query 2 : ?-
odd_number(7).Output : No.

Explanation : As our knowledge base does not contain the above fact, the output was 'No'.

Advantages :

1. Easy to build databases. Doesn’t need a lot of programming effort.
2. Pattern matching is easy. Search is recursion based.
3. It has built in list handling. Makes it easier to play with any algorithminvolving lists.

Disadvantages :

1. LISP (another logic programming language) dominates over prolog withrespect to I/O features.
2. Sometimes input and output is not easy.

Applications :

Prolog is highly used in artificial intelligence(AI). Prolog is also used for patternmatching over natural language parse trees.

Programs

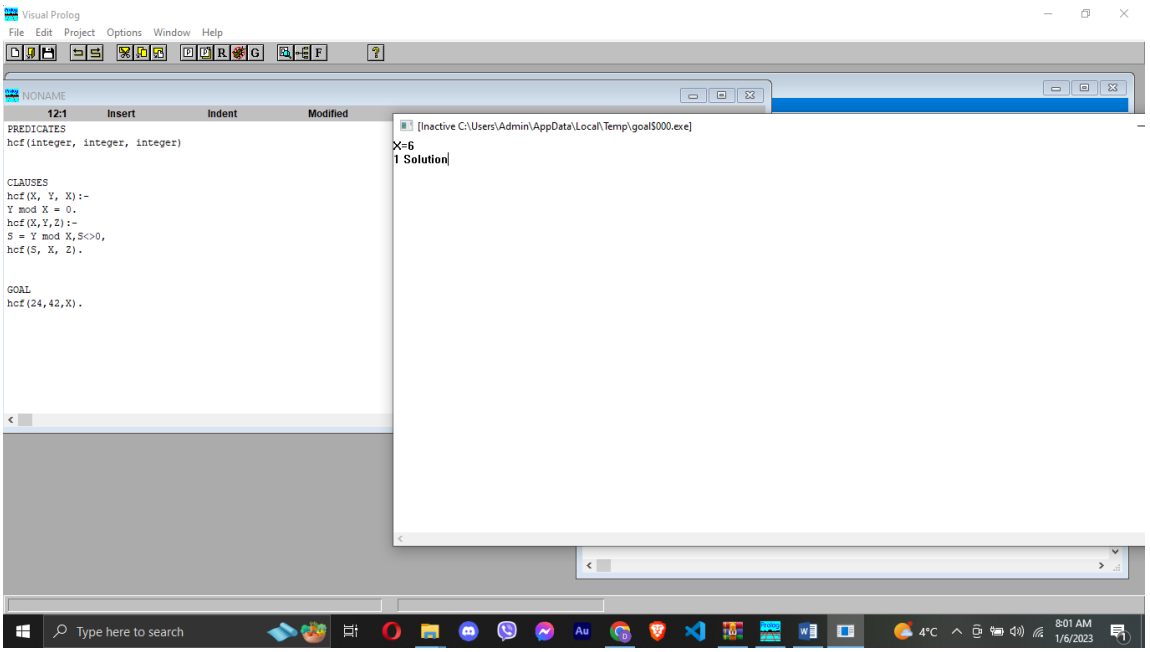
1. Write a program to find the hcf of two numbers.

```
PREDICATES
hcf(integer, integer, integer)


CLAUSES
hcf(X, Y, X):-
Y mod X = 0.
hcf(X,Y,Z):-
S = Y mod X,S<>0,
hcf(S, X, Z).


GOAL
hcf(24,42,X).
```

OUTPUT:



Description:

In this problem, we find the hcf of the numbers. The goal is quite straight forward. For the solution, firstly it is checked whether one of the number is a common factor for the both of them.

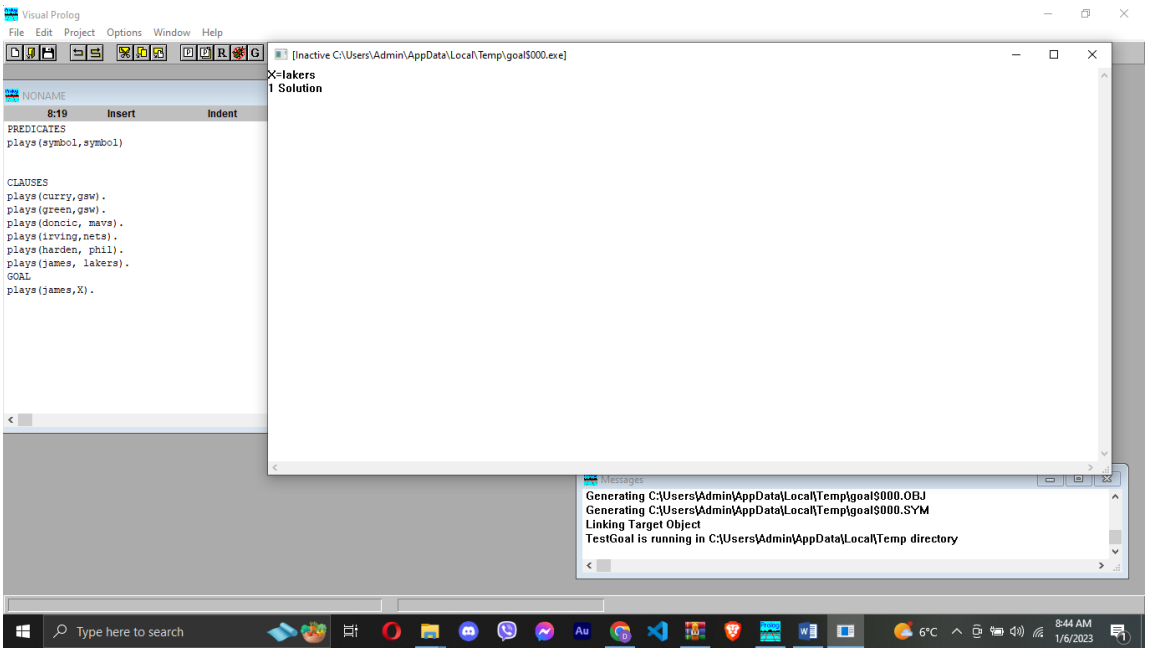
2. Write a program of your choice. Give some facts and use some rules tomake a few deductions

PREDICATES
plays(symbol,symbol)

CLAUSES
plays(curry,gsw).
plays(green,gsw).
plays(doncic, mavs).
plays(irving,nets).
plays(harden, phil).
plays(james, lakers).

GOAL
plays(james,X).

OUTPUT:



Description:

Here, we used some facts and rules and made the deduction that james plays for lakers.

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

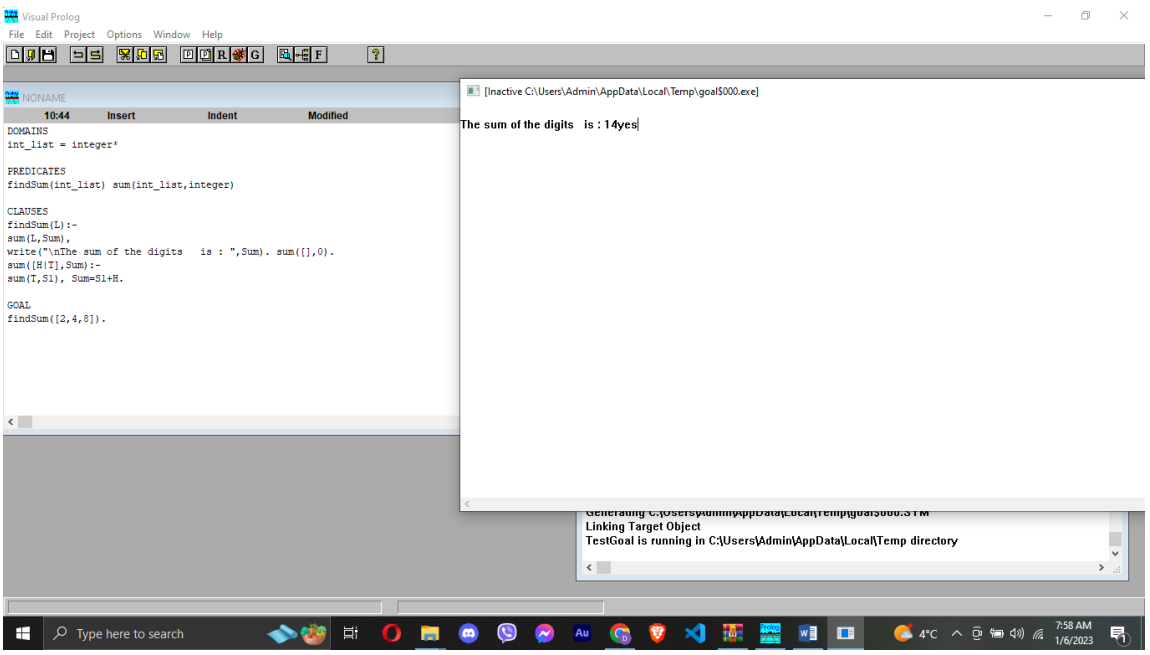
```
DOMAINS
int_list = integer*

PREDICATES
findSum(int_list) sum(int_list,integer)

CLAUSES
findSum(L):-
sum(L,Sum),
write("\nThe sum of the digits   is : ",Sum). sum([],0).
sum([H|T],Sum):-
sum(T,S1), Sum=S1+H.

GOAL
findSum([2,4,8]).
```

OUTPUT:



Description:

In this problem, we find the sum of the contents of the list. Here, we used the fact that a list is divided into a head element and tail list to recursively add the elements to the sum.

4. Write a program to find the length of a list.

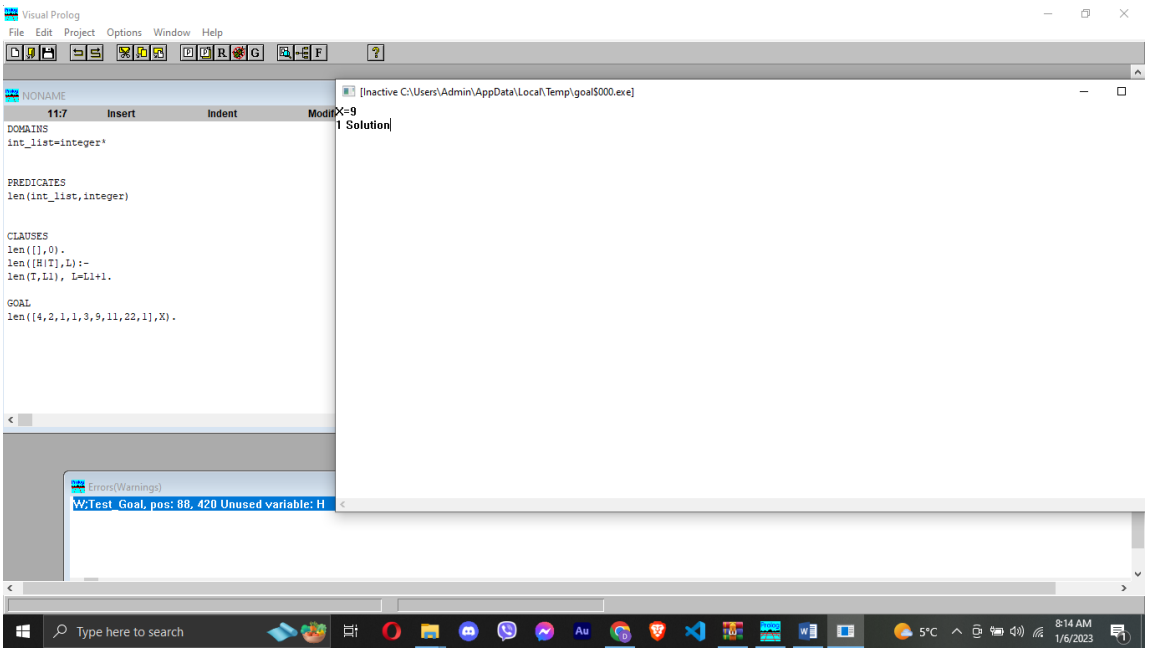
```
DOMAINS
int_list=integer*


PREDICATES
len(int_list,integer)


CLAUSES
len([],0).
len([H|T],L):-
len(T,L1), L=L1+1.

GOAL
len([4,2,1,1,3,9,11,22,1],X).
```

OUTPUT:



Description:

Here, we find the length of the given list. By popping the head elements recursively from the tail lists in the list, and increase the count L by 1 every time. The final value gives the length of the list.

5. Write a program to append two lists.

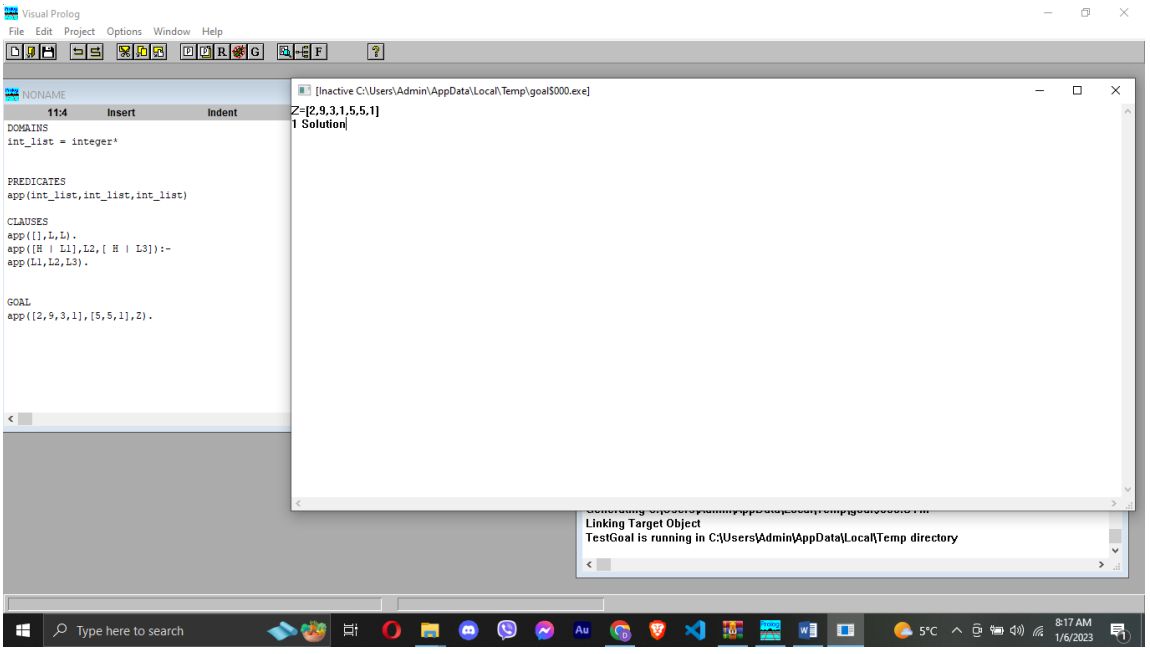
```
DOMAINS
int_list = integer*


PREDICATES
app(int_list,int_list,int_list)

CLAUSES
app([],L,L).
app([H | L1],L2,[ H | L3]):-
app(L1,L2,L3).


GOAL
app([2,9,3,1],[5,5,1],Z).
```

OUTPUT:



Description:

In this program, we appended the given lists. Here too, we use the head and tail entities, and join the tail of the first list with the second list.

6. Write a program which takes a list of integers and displays only 1s and 2s. (If the input is [1,2,4,5,2,4,5,1,1] the solution list should be [1,2,2,1,1].)

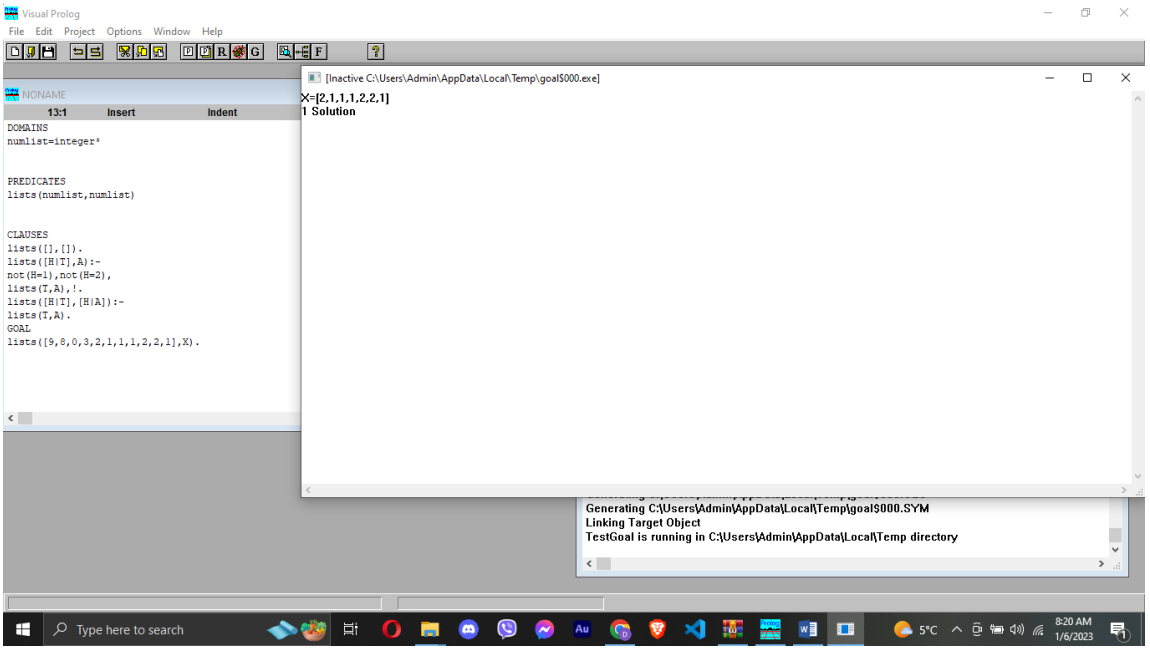
```
DOMAINS
numlist=integer*

PREDICATES
lists(numlist,numlist)

CLAUSES
lists([],[]).
lists([H|T],A):-
not(H=1),not(H=2),
lists(T,A),!.
lists([H|T],[H|A]):-
lists(T,A).

GOAL
    lists([9,8,0,3,2,1,1,1,2,2,1],X).
```

OUTPUT:


Visual Prolog
File Edit Project Options Window Help
NONAME
13:1 Insert Indent
DOMAINS
numlist=integer*
PREDICATES
lists(numlist,numlist)
CLAUSES
lists([],[]).
lists([H|T],A):-
not(H=1),not(H=2),
lists(T,A),!.
lists([H|T],[H|A]):-
lists(T,A).
GOAL
lists([9,8,0,3,2,1,1,1,2,2,1],X).
[Inactive C:\Users\Admin\AppData\Local\Temp\goal$000.exe]
X=[2,1,1,1,2,2,1]
1 Solution
Generating C:\Users\Admin\AppData\Local\Temp\goal$000.SYM
Linking Target Object
TestGoal is running in C:\Users\Admin\AppData\Local\Temp directory
Type here to search
5°C
8:20 AM
1/6/2023

Description:

Here, we filter out the 1s and 2s in the list. By traversing through the list and checking whether the given element is a 1 or a 2 or any other number, we achieve this feat.

7. Write a program to delete a given item from a list.

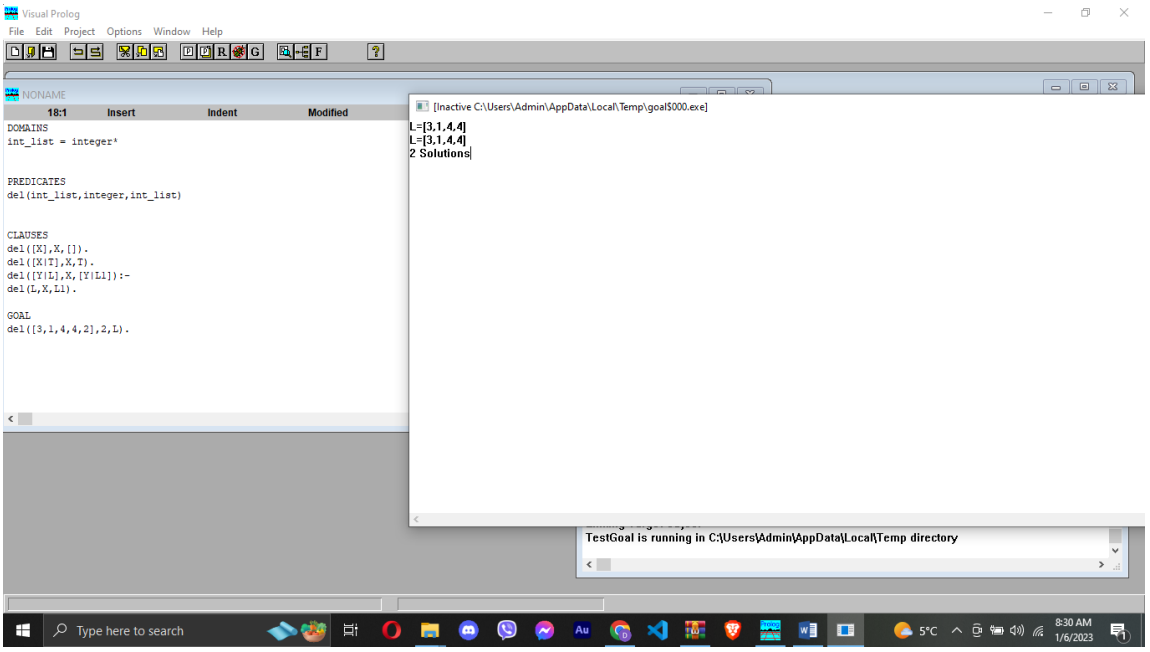
```
DOMAINS
int_list = integer*

PREDICATES
del(int_list,integer,int_list)

CLAUSES
del([X],X,[]).
del([X|T],X,T).
del([Y|L],X,[Y|L1]):-
del(L,X,L1).

GOAL
    del([3,1,4,4,2],2,L).
```

OUTPUT:


Visual Prolog
File Edit Project Options Window Help
NONAME
18:1 Insert Indent Modified
DOMAINS
int_list = integer*
PREDICATES
del(int_list,integer,int_list)
CLAUSES
del([X],X,[]).
del([X|T],X,T).
del([Y|L],X,[Y|Ll]):-
del(L,X,Ll).
GOAL
del([3,1,4,4,2],2,L).
[Inactive C:\Users\Admin\AppData\Local\Temp\goal$000.exe]
L=[3,1,4,4]
L=[3,1,4,4]
2 Solutions
TestGoal is running in C:\Users\Admin\AppData\Local\Temp directory
Type here to search
5°C
8:30 AM
1/6/2023

Description:

Here, we delete an item from the given list. First the item is found by traversing the list as done in the previous question. Then, the element is deleted and the final list is printed.

Discussion

In this lab, we dealt with the basics of Prolog. A few programming exercise were done to get a better understanding. Things like facts and predicates were also introduced. Programs for finding hcf between the given numbers, finding the length of a given list, appending two lists, deleting an element of a list, etc. were coded in Visual Prolog. The overall programming involved functions like predicates, clauses, goals and lists.

Conclusion

Thus, we learned basics of Prolog and also learned some basic concepts by programming in Visual Prolog.