# KADI SARVA VISHWAVIDYALAYA

## LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH

## GANDHINAGAR



**Department of**

**Information Technology**

**Subject: Artificial Intelligence (*CT601*-N)**

# Laboratory Manual

Prepared By:

**Samarth Mistry**

**19BEIT30060**

**6IT**

# LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

DEPARTMENT OF
INFORMATION TECHNOLOGY


# *CERTIFICATE*

Mr. **Mistry Samarth Alkeshkumar** of 6th IT Enrolment No. **19BEIT30060** Exam No, _____ has satisfactorily completed his/her term work in ***Artificial Intelligence (CT601-N)*** for the term ending in **Dec-May-2022**.

Date: _____

_____ Dr. Mehul Barot

**Subject Coordinator** **HOD-IT**

# PRACTICAL 1

AI practical #1

PAGE:
DATE: / /

Aim: To study of facts, objects, Predicates Variables, Rule and Unification in PROLOG

> PROLOG (Programing in Logic)
It is a declarative programming language like SQL which has an importance role in the field of Artifical Intelligence.

The primary focus in the PROLOG is the logic which is applied to perform any task. The logic is expressed with the help of some rules & facts. It is an Object oriented programming language.

The organized data has is known as facts. The predefined constraints & conditions to perform any task are known as rule. The queries are executed to check if the task is accomplished properly or ~~root~~ not.

PROLOG has four parts:

(i) Editor - Editor is used to write the programming logic by defining variables, facts as objects and rules.

(ii) Dialog - Dialog is used to execute the program Here the essential queries can be executed to check if logic is implement

is correct or not.

(iii) Message - This blocks shows the status of action performed in code like saving the file, compilation was successful or not, etc

(iv) Trace : Trace is used to debug the program. It is used to find any errors which are not easily traceable.

(v) Facts - Facts can be described as symbolic relationships.

Example : Bob is a student
-> student(BOB):
Factual expression is called a clause.

(vi) Object : Object is the name of element of certain type. Object could be as follows

- char : single character
- Integer : Integer values
- Real : Floating point numbers
- String : Sequence of characters
- Symbol : Character sequence of left numbers & underscores with characters as lowercase letters.
- File : Symbolic name.

ex Bob is a student
T(object)

- Relation : A relation is a name that defines the way in which a collection of object or variables belong ~~wrt~~ together.

- Predicate: It is a fact without a period at the end.

ex. Car is Blue.
is(Car, Blue) is not a fact, so it is a predicate.

The elements within parenthesis are arguments to predicate, which may be object or variables.

- Variables : In prolog, variables means unknown or undefine quantity. There are three types of variable.

(i) Bound variables :
- Bound means initiated or not defined at particular time.
- A variable which is Bound with a value at a particular given time.
- Value is been assigned to variable than it is called bound variable.

(ii) Free variables :
These variables which are not initiated or defined at a particular given time is called.

free variable.

(iii) Anonymous variables:
- It means no variable & no value

- UNIFICATION: Unification is a pattern matching process. The process by which a prolog tries to match a term against the facts in an efforts to prove a goal is called unification

- Predicates unify with each other if:
(i) They have same relation name
(ii) They have same number of arguments
(iii) All arguments pairs unify with each other

- Rules: A Rule is an expression that shows that the truth of a particular fact depends upon one or more facts. A Rule express a relationship between facts.

- Prolog expression are comprised of following truth functional symobals.

| English | Predicate | Prolog |
|---|---|---|
| AND | $\wedge$ | , |
| OR | $\vee$ | ; |
| IF | $\rightarrow$ | :- |
| NOT | $\sim$ | not |

- A Rule is a predicate expression that uses logical implementation (:-) to describe a relationship among facts.

Thus a prolog rule takes the form
left hand side :- right hand side

This is interpreted as
left hand side if right hand side

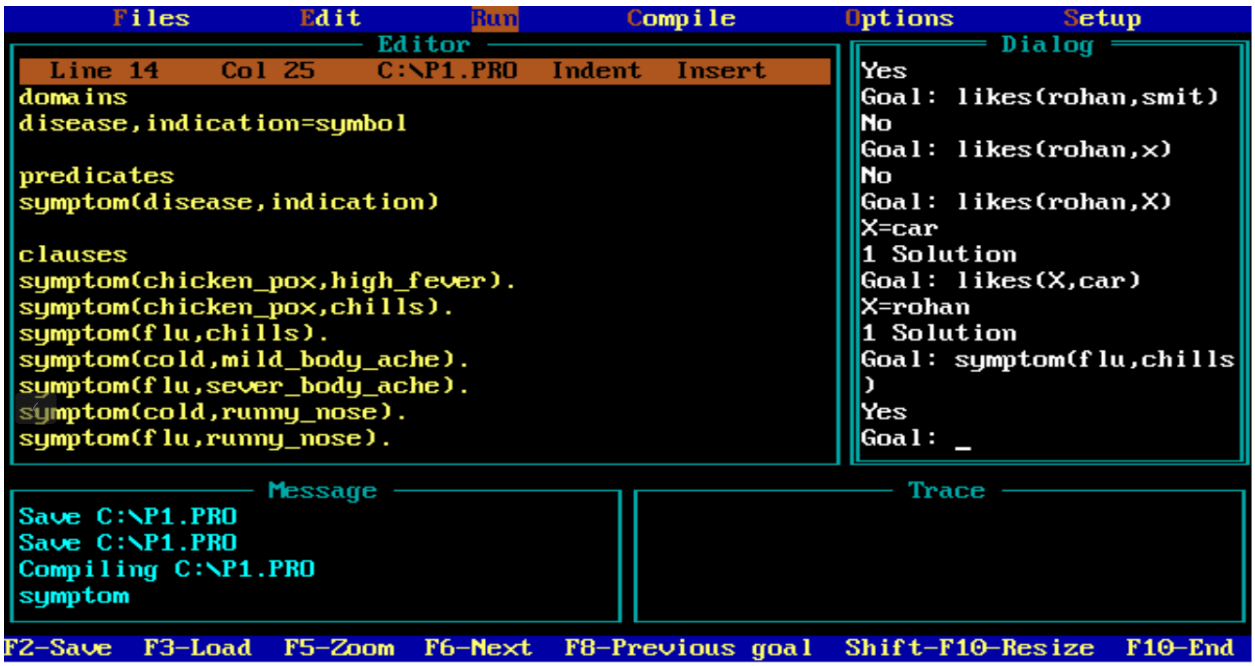
It is restricted to a single positive literal which means it must consist of a positive atomic expression. It can't be neglected and it can't contain logical connectives.

This notation is known as Horn clause.

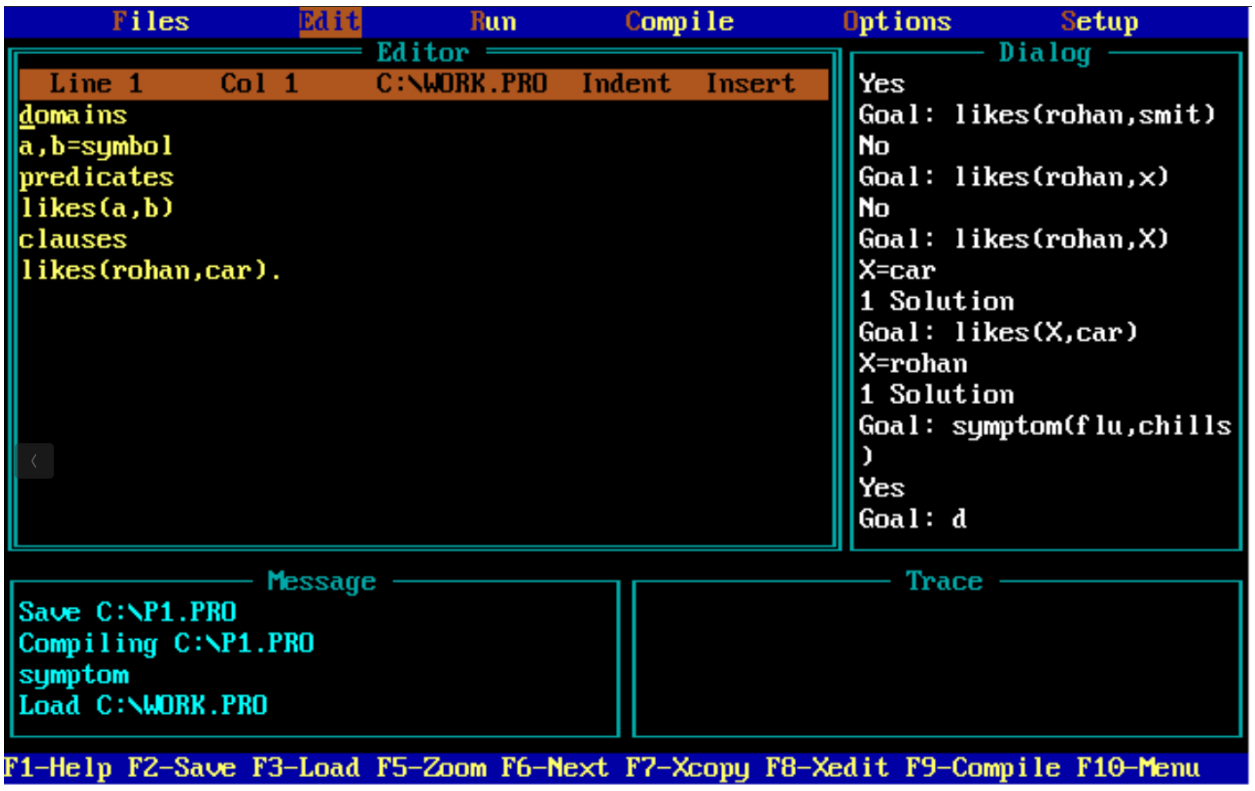
- In Horn clauses logic, the left hand side of clause is conclusion & must be single positive literal. The right hand side contains the premises. The Horn clause calculus is equivalent to first order predicate calculus.

Part-1: To implement a program for likes.

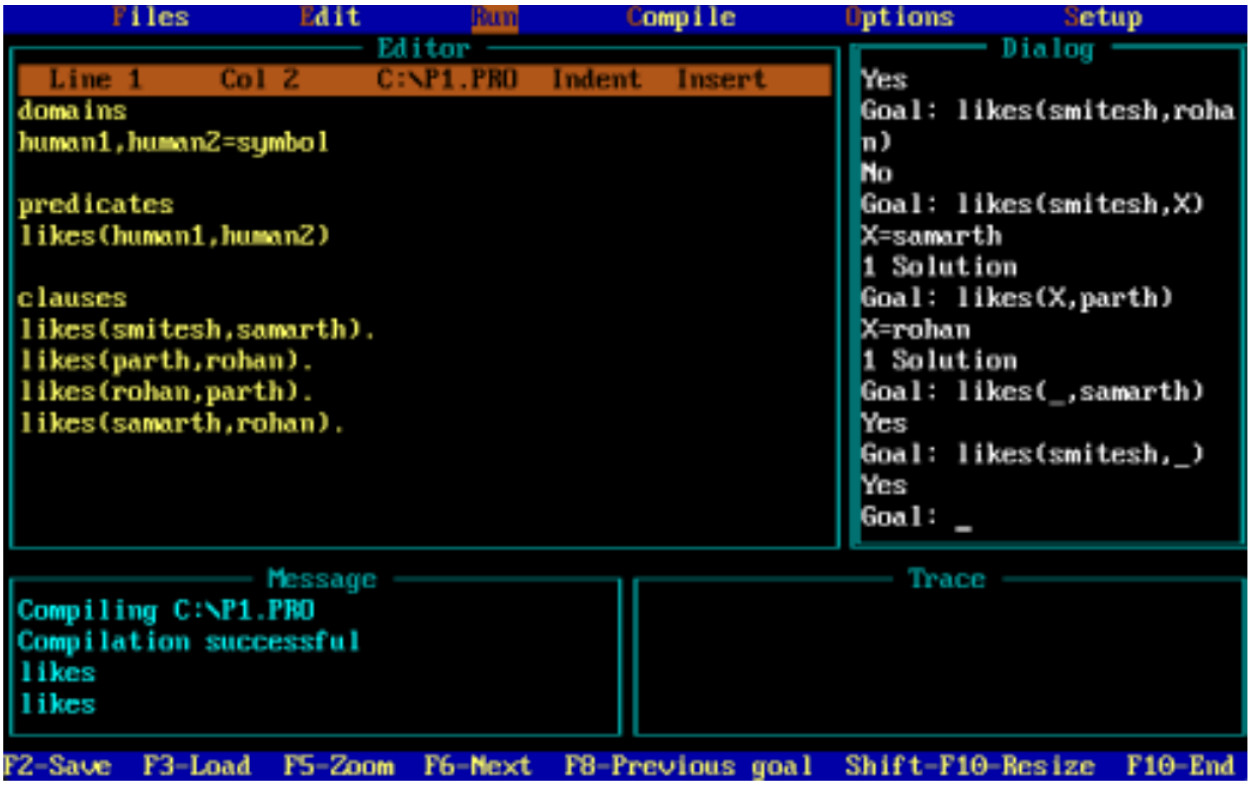
Use of Bound Variables:




Use of Free Variables :

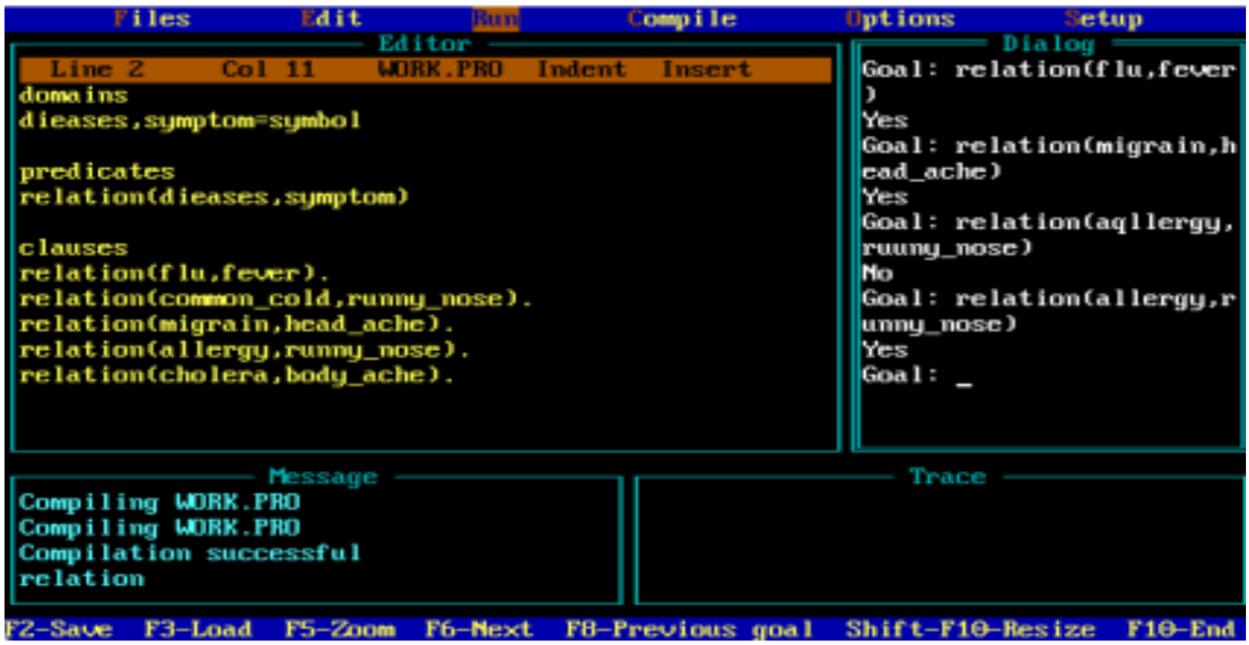


Use of Anonymous Variables:

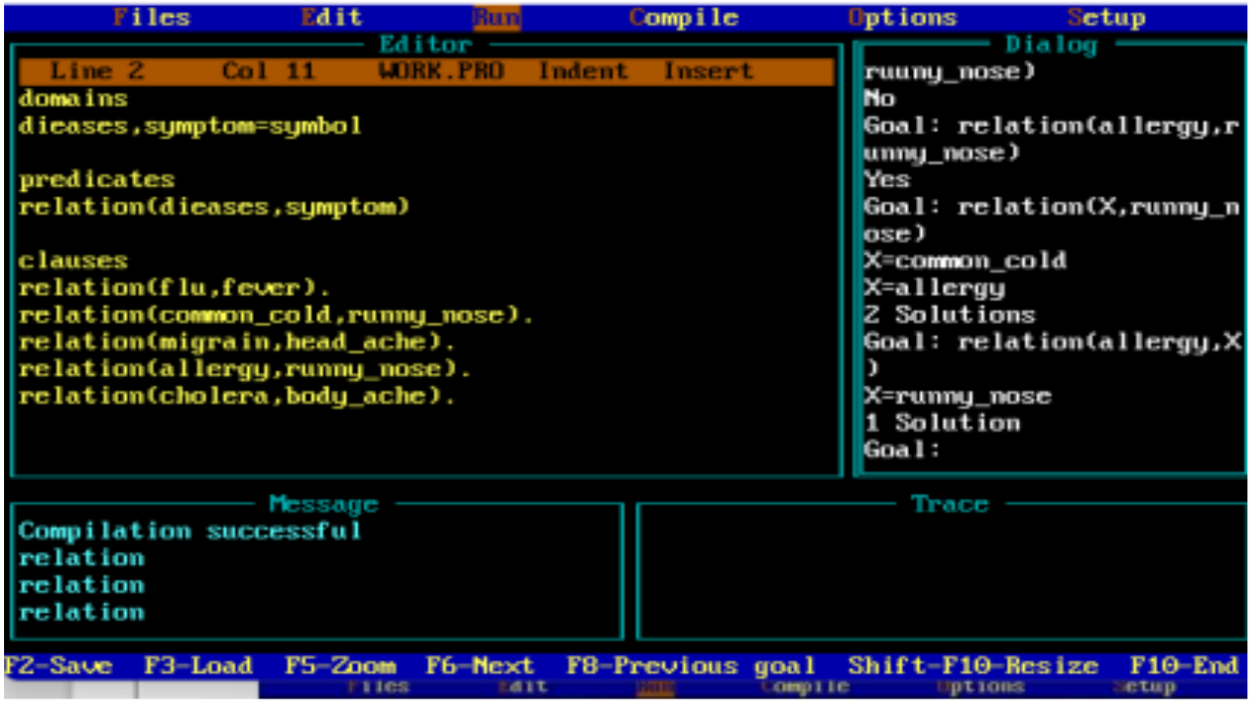


**Part 2**: To implement a program of Diseases and it's Symptoms.

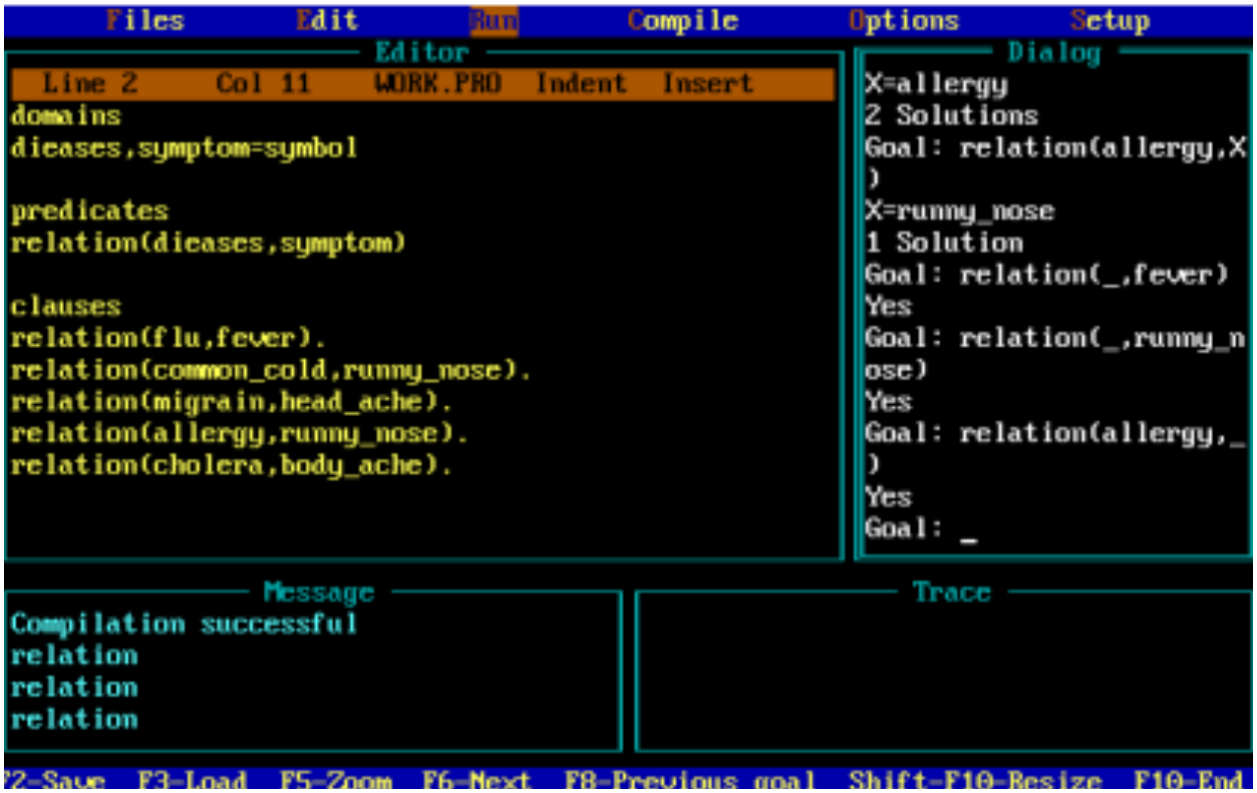
Use of Bound Variables:




Use of Free Variables:



Use of Anonymous Variables:



Part 3 : Implement the following knowledge with the statement.

Knowledge:

(a.) Burger is a Food

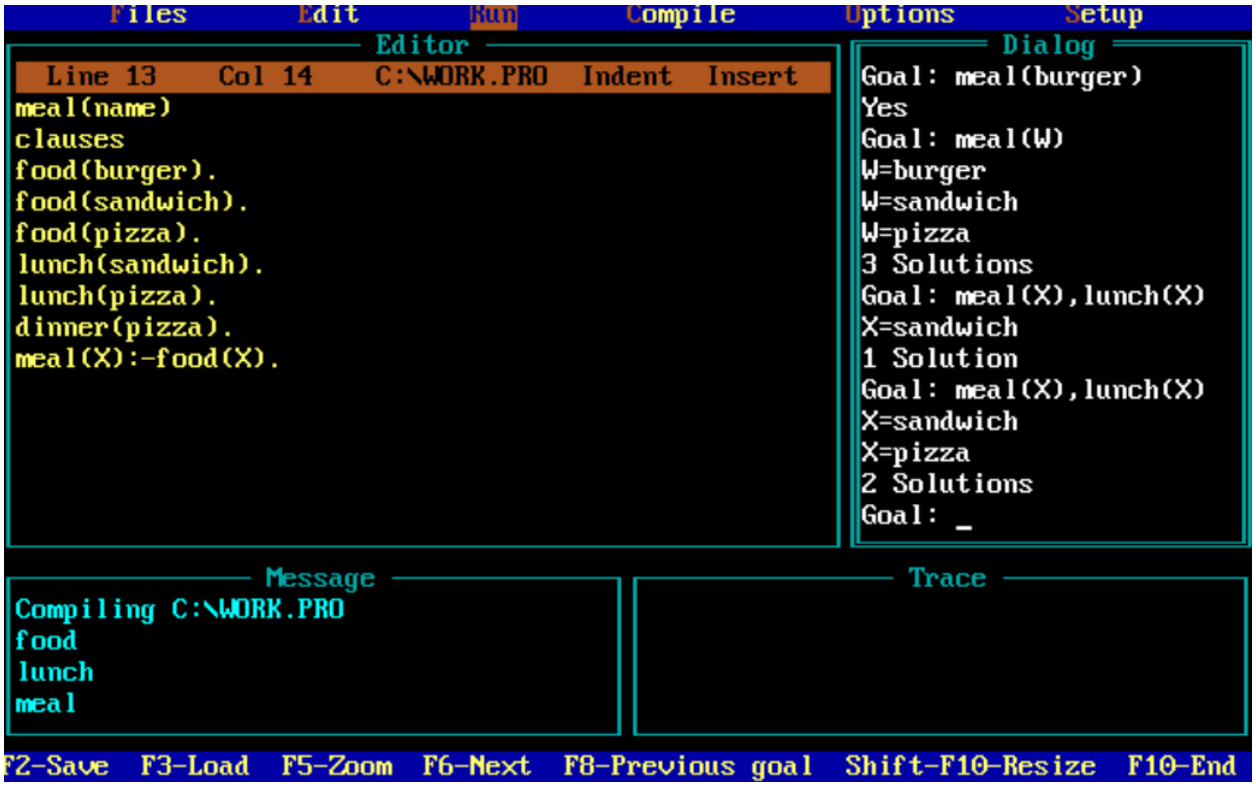
(b.) Sandwich is a Food.

(c.) Pizza is a Food.

(d.) Sandwich is a Lunch.

(e.) Pizza is a Dinner.

Statement: Anything is a meal, if it is a food.



Part 4: Implement the following knowledge with the statement.

Knowledge:

(a) A studies Java

(b) B studies Java

(c) C studies Python

(d) B studies DS

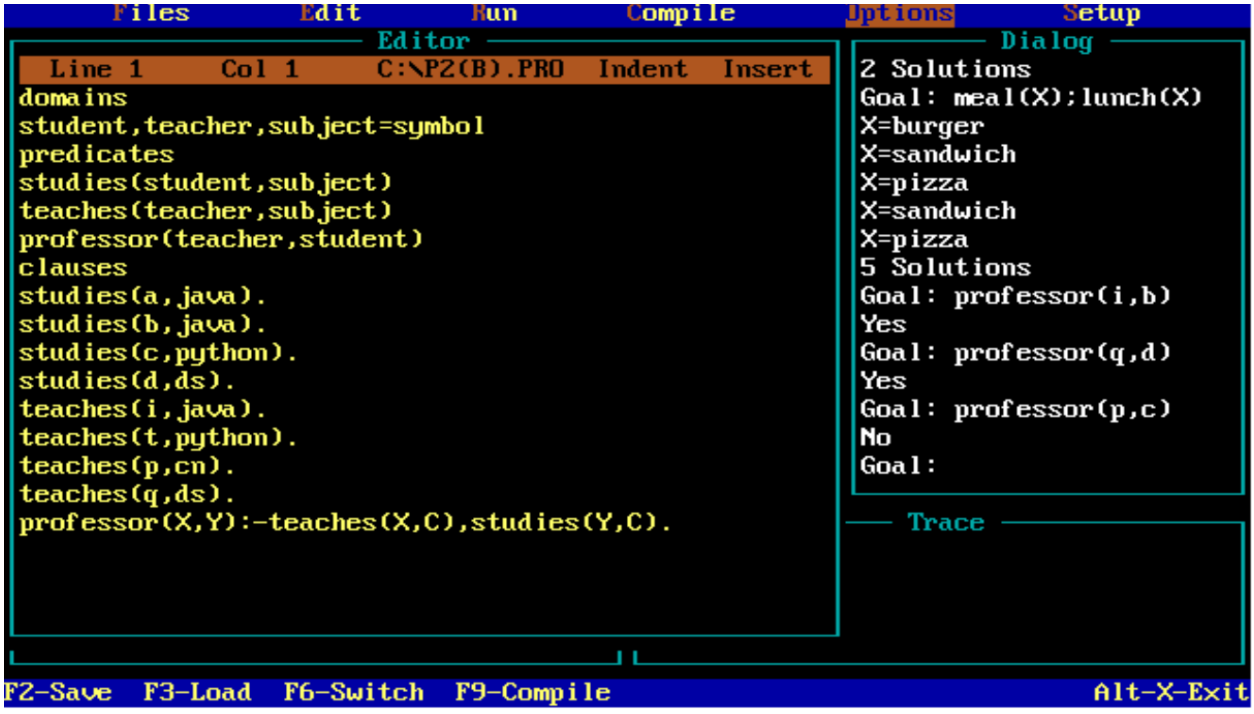
(e) I teaches Java

(f) T teaches Python

(g) T teaches CNS

(h) Q teaches DS

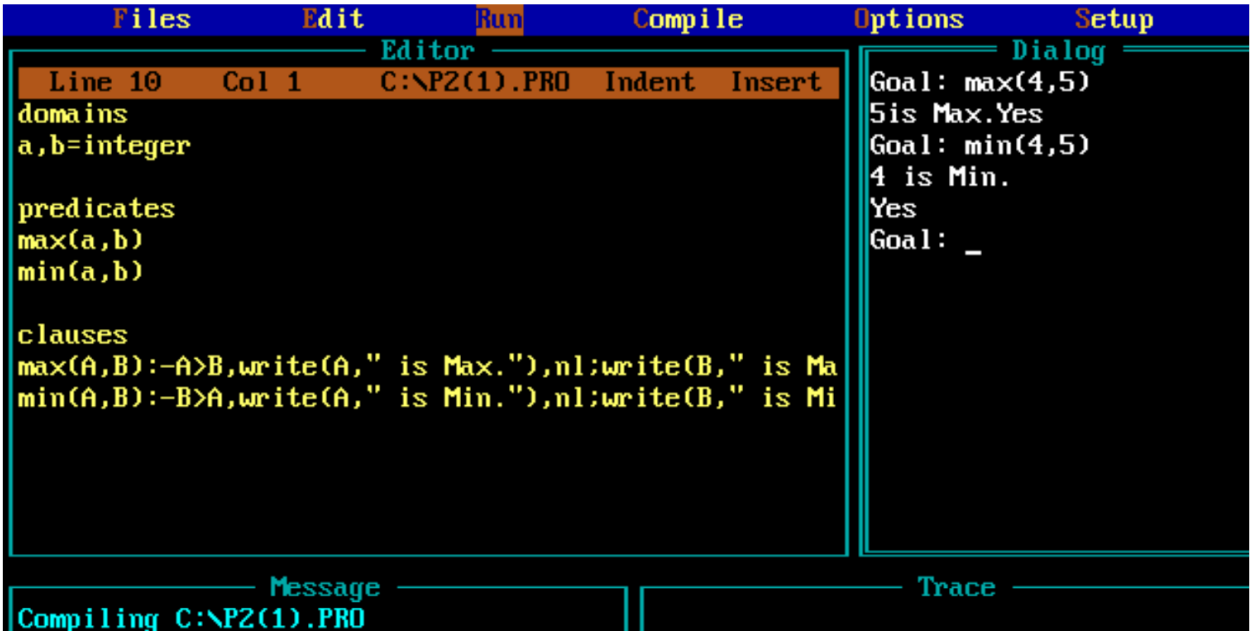
Statement: X is a professor of Y if it X teaches C and Y studies C.



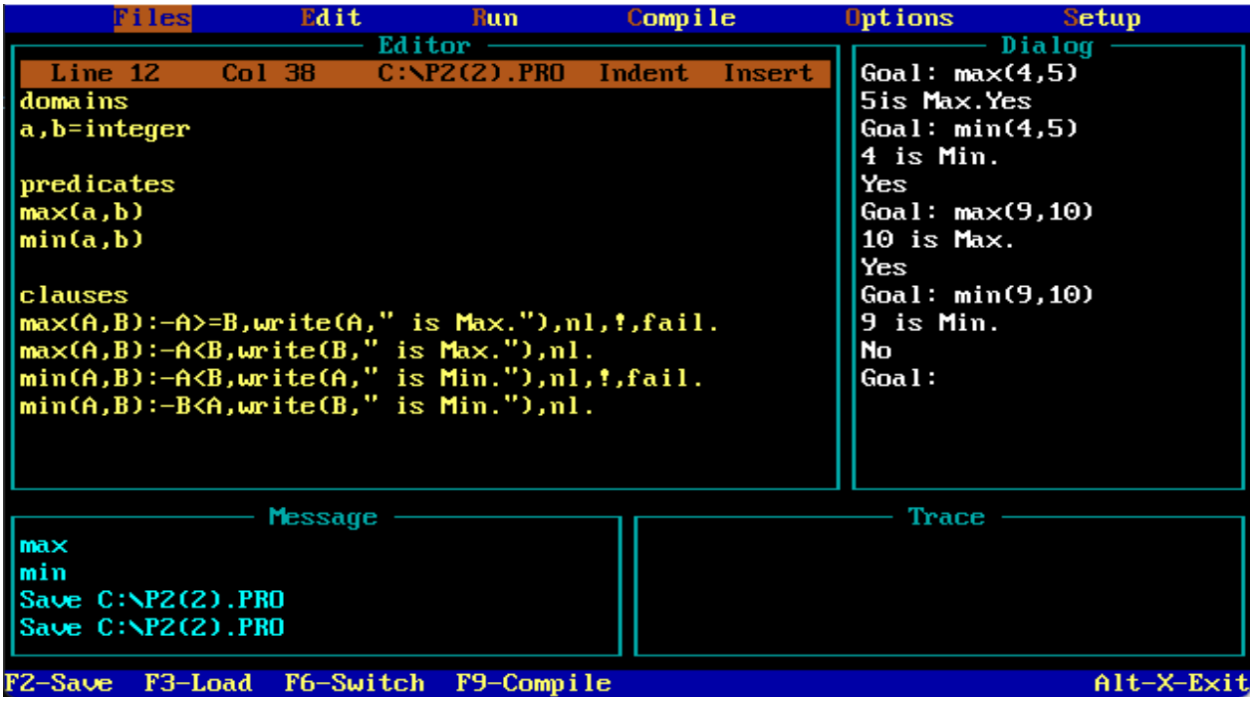
# PRACTICAL :2

**AIM:** Write a program to implement the “cut” and “fail” predicate in PROLOG.

**Part 1**: Implement program to find Minimum and Maximum number without using cut and fail predicates in prolog.



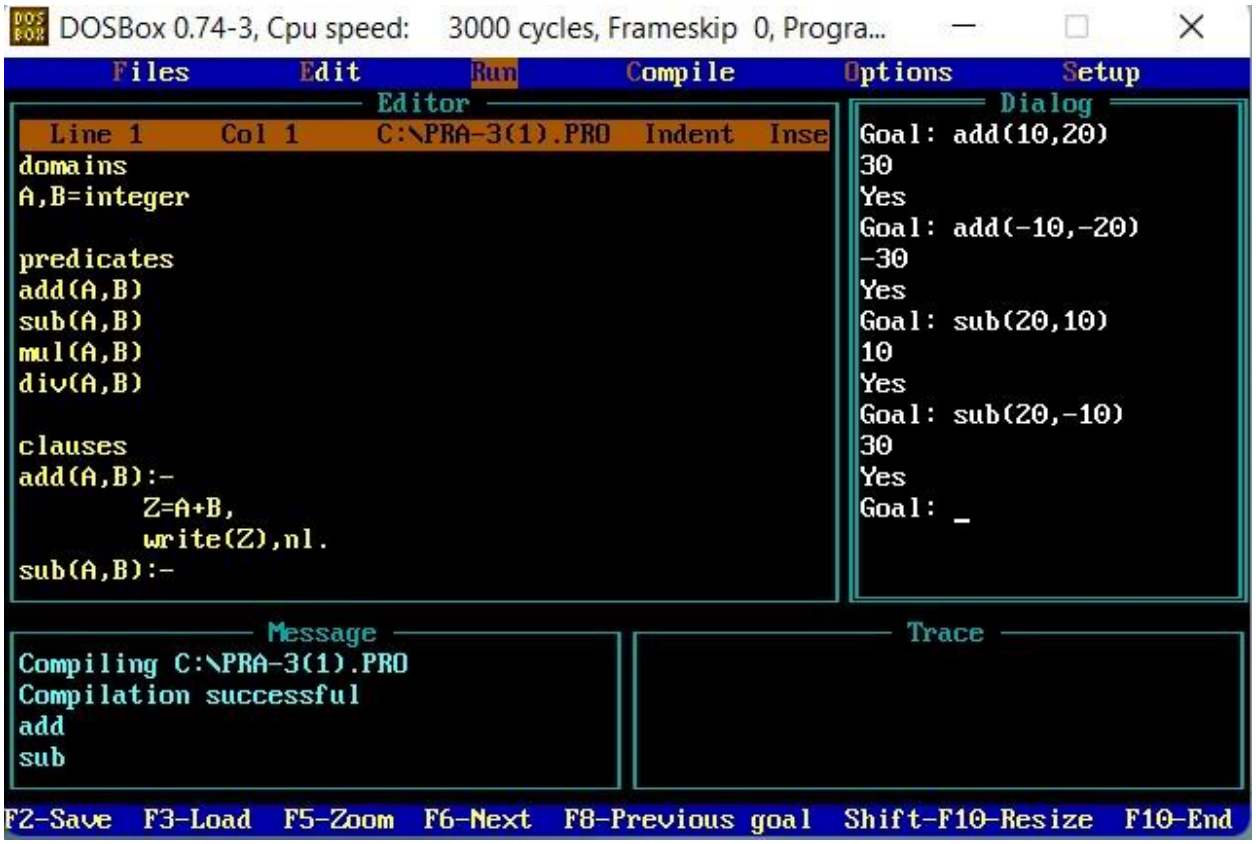

**Part 2**: Write a program to find Minimum and Maximum using cut and fail predicates

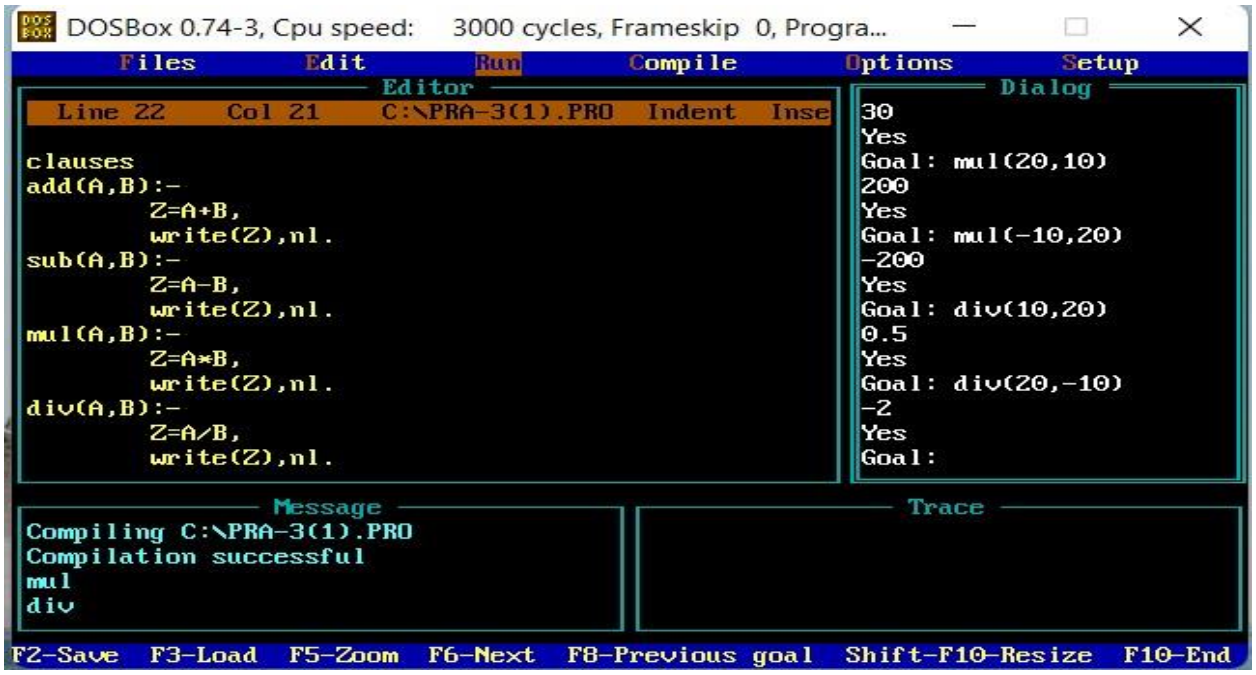


# PRACTICAL : 3

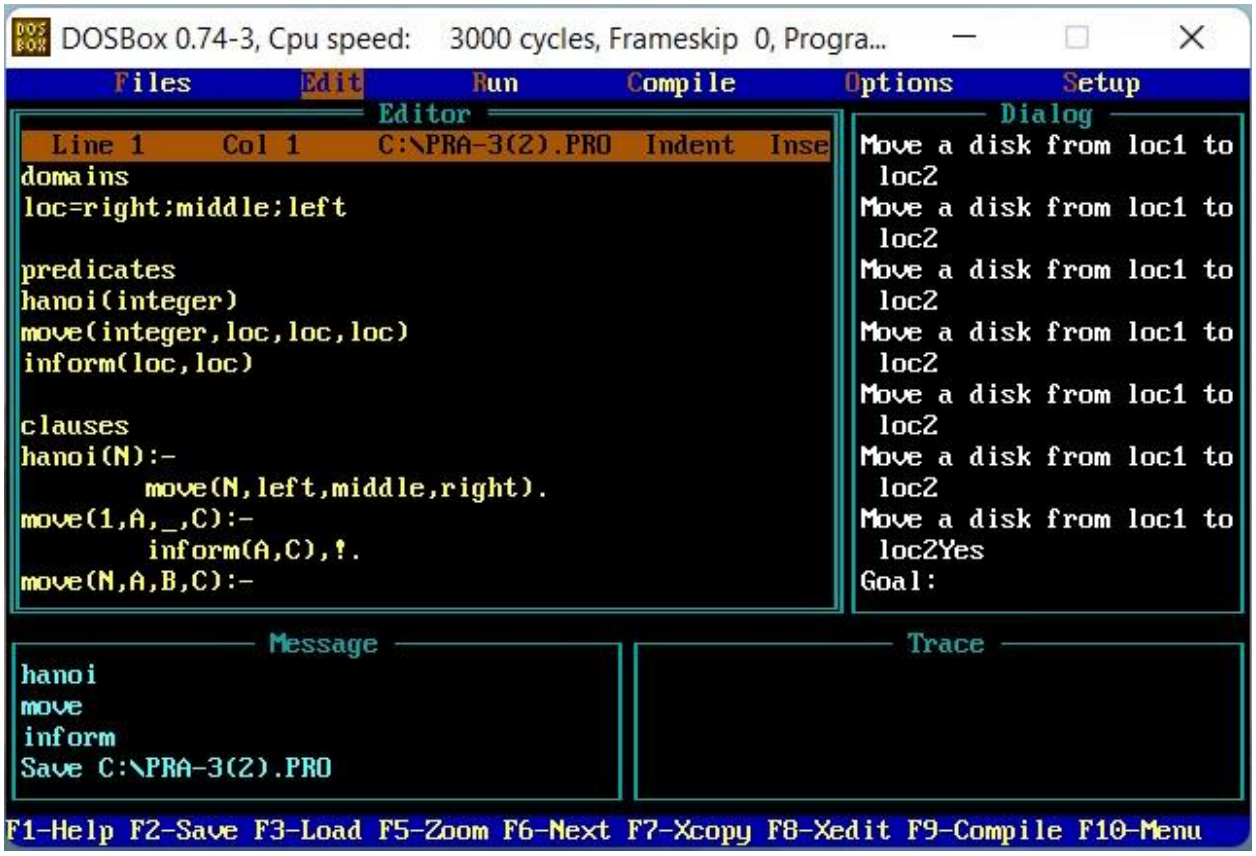
**AIM:** Write a program to implement arithmetic operators, simple input/output and compound goals in PROLOG.

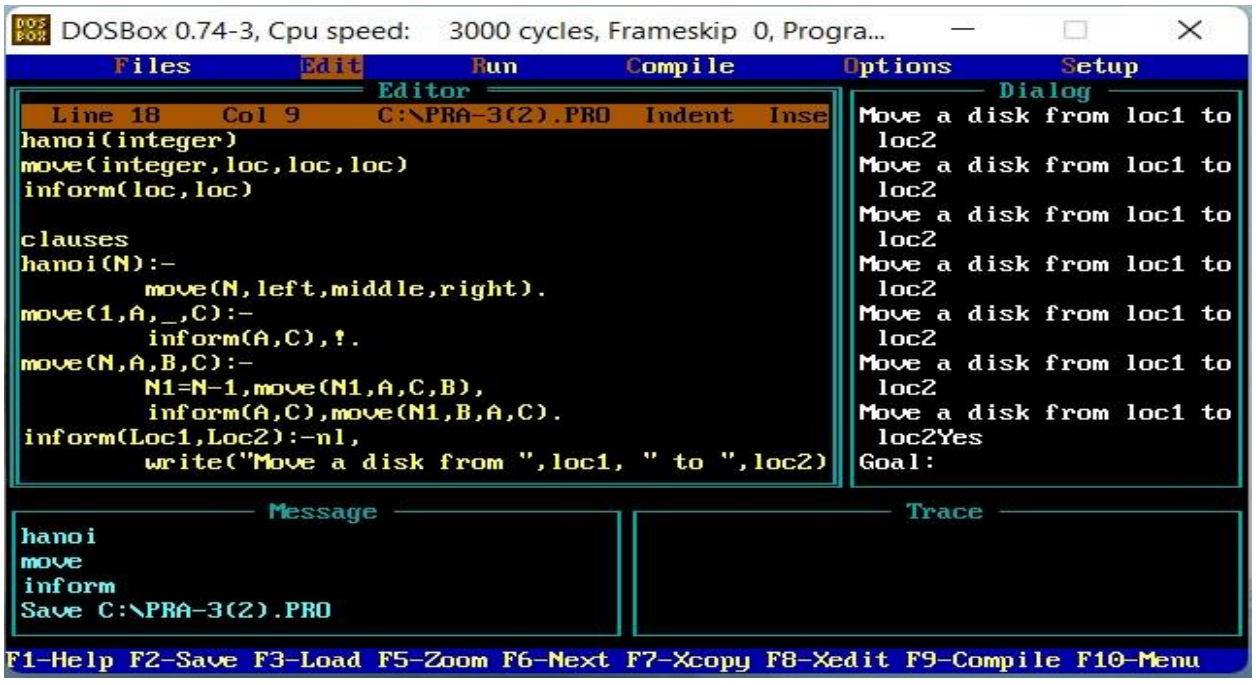
**Program 1:** Program to implement arithmetic operators.



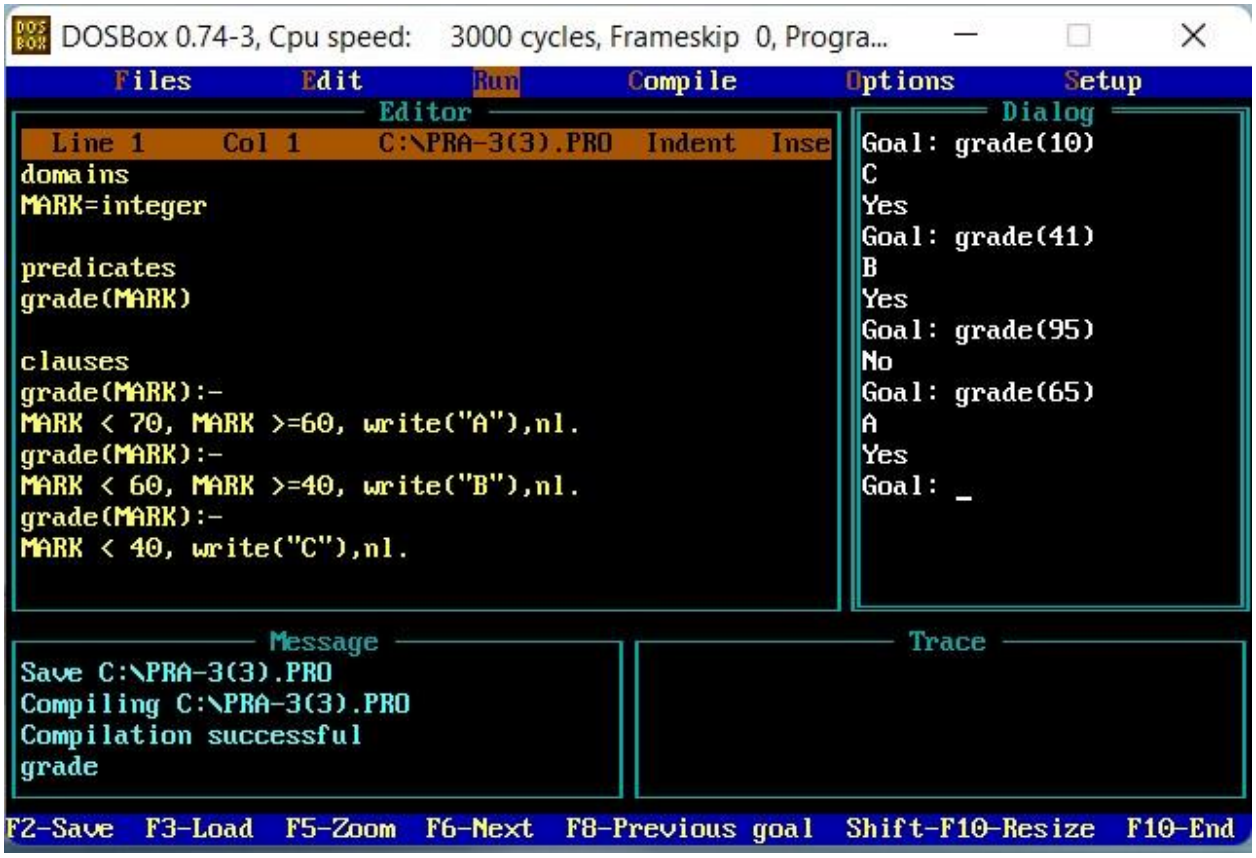



**Program 2 :** Program to implement tower of hanoi .

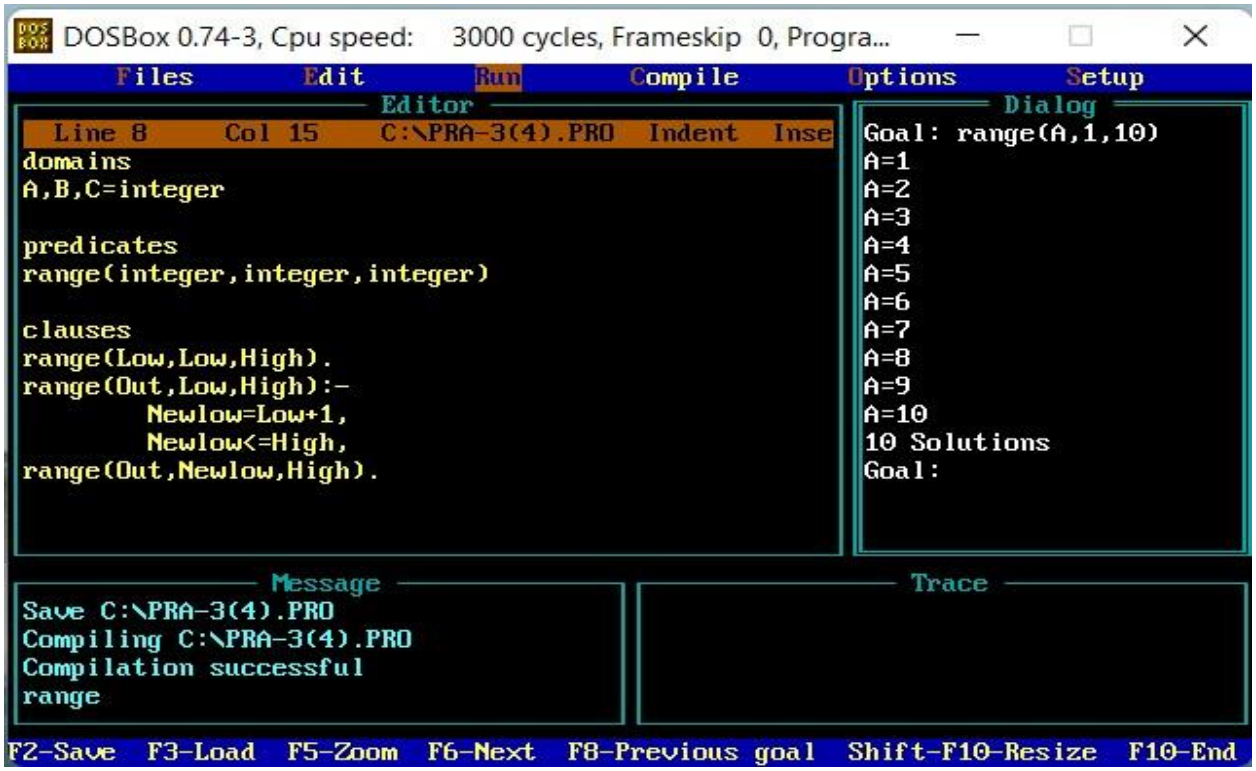


**Program 3 :** Program to find grade .

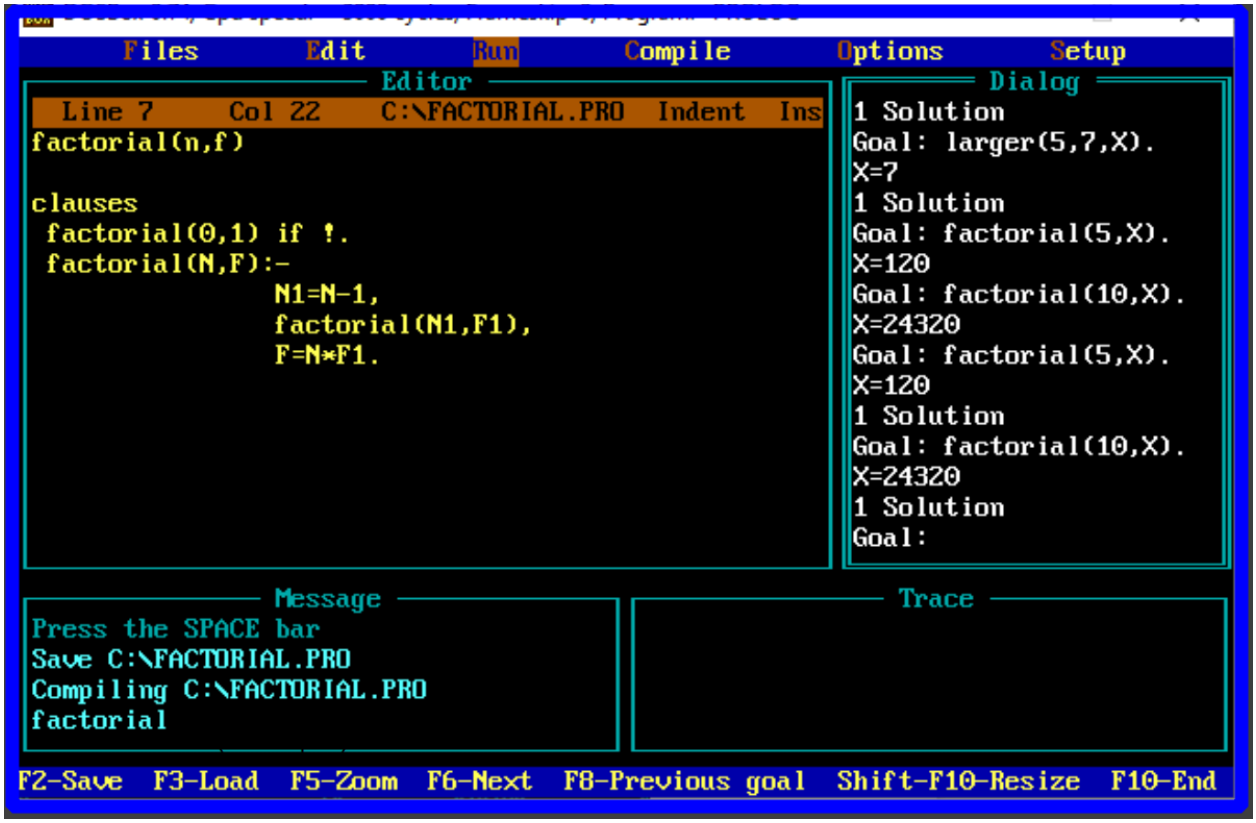



**Program 4 :** Program to print the values between the given range.



## PRACTICAL 4

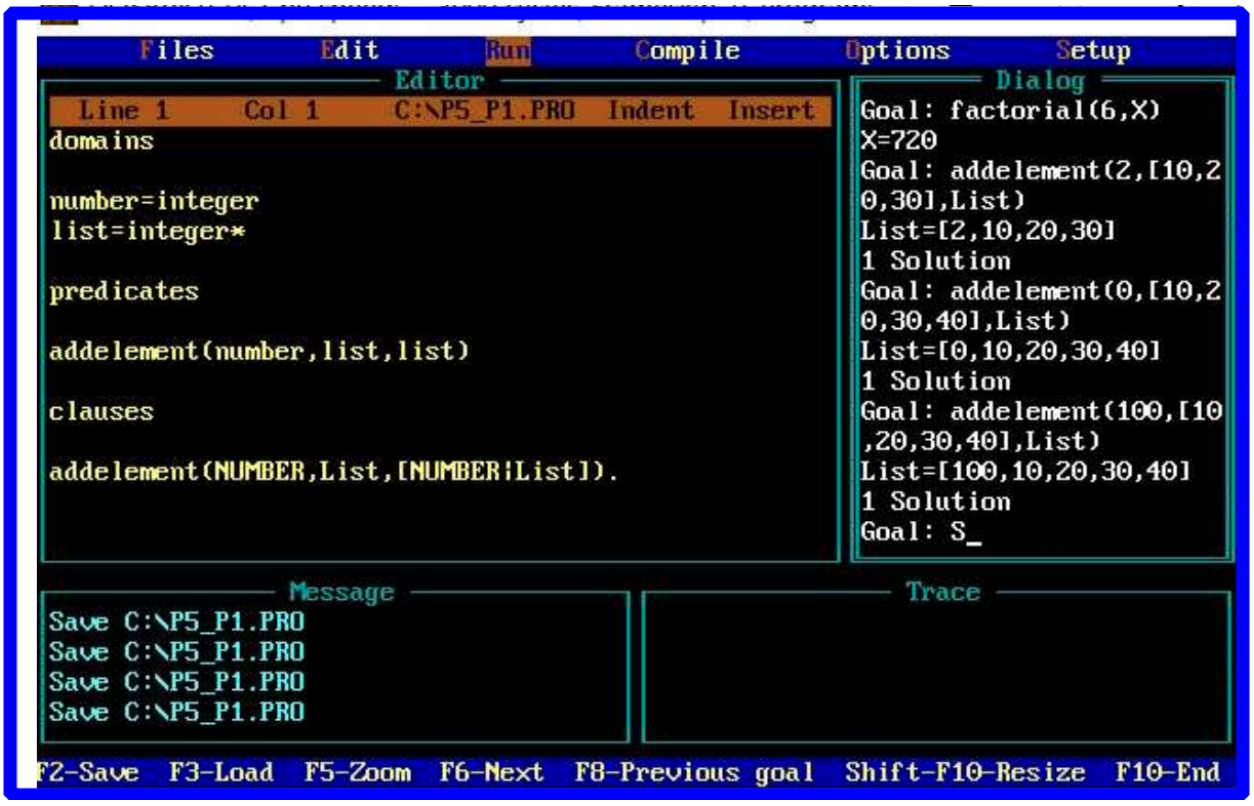
**AIM:** Write a program to implement recursion in PROLOG. Program to implement factorial.



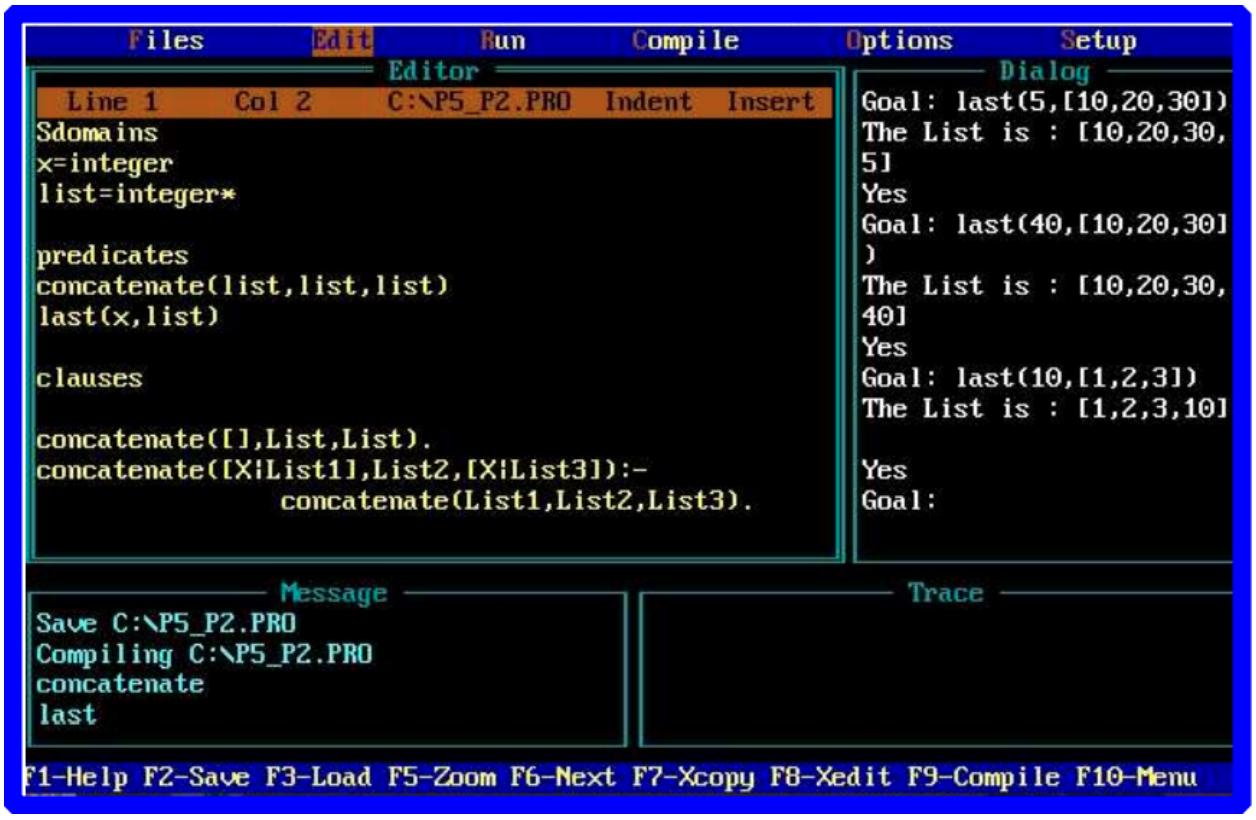
# PRACTICAL 5

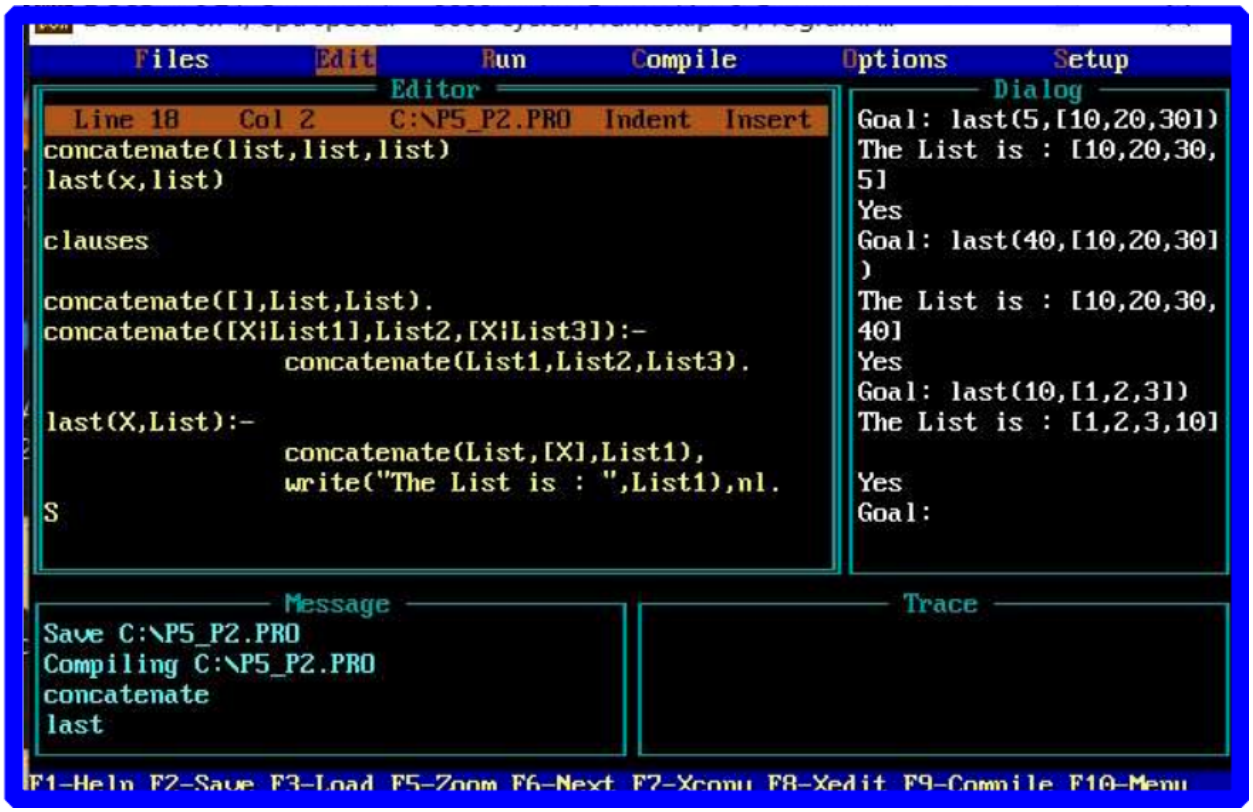
**AIM:** Write a program to implement Lists in PROLOG.

**Part 1**: Implementing program to add an element to the List in PROLOG.



**Part 2**: The PROLOG program defines the relation last(item,list) so that the item is the last element of the list using concatenation.



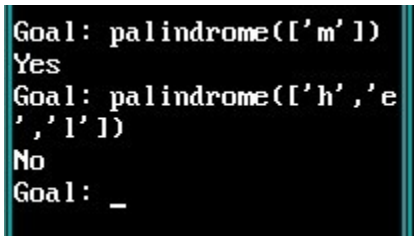



## PRACTICAL : 6

**Program (1):** Code for Prolog program to check whether a given word is a palindrome or not in Artificial Intelligence.

```
domains
        x = char
        l = char*
predicates
        palindrome(l)
        reverse(l,l)
        concatenate(l,l,l)
clauses
concatenate([],List,List).
        concatenate([X|List1],List2,[X|List3]) :-
                concatenate(List1,List2,List3).
reverse([],[]).
reverse([X|Tail],List) :-
reverse(Tail,Tail1),
concatenate(Tail1,[X],List).
palindrome(List):-
        reverse(List,List).
```

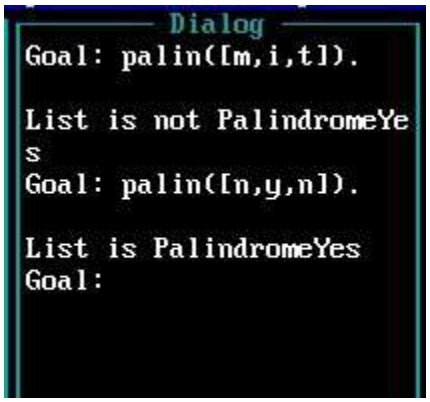
**OUTPUT :**



**Program (2):** Code for Prolog program to check whether a given list is palindrome or not in Artificial Intelligence

```
Domains
        list=symbol*
predicates
palin(list)
findrev(list,list,list)
compare(list,list)
clauses
palin(List1):-
        findrev(List1,[],List2),
        compare(List1,List2).
findrev([],List1,List1).
findrev([X|Tail],List1,List2):-
        findrev(Tail,[X|List1],List2).
        compare([],[]):-
                write("\nList is Palindrome").
compare([X|List1],[X|List2]):-
        compare(List1,List2).
compare([X|List1],[Y|List2]):-
        write("\nList is not Palindrome").
```

**OUTPUT :**

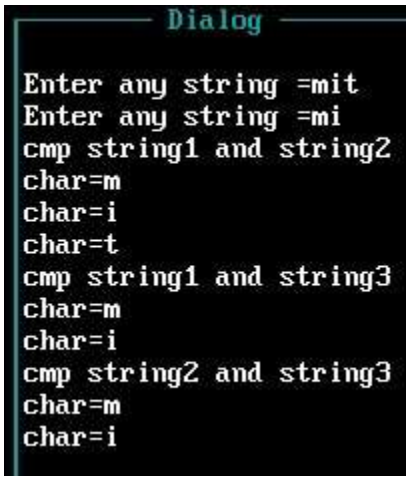


**Program (3):** Code for Prolog program to compare characters, strings and also reverse string in Artificial Intelligence

```
domains
strlist = string*
predicates
start
        createlist(integer,strlist,strlist,strlist)
        strcmp(string,string)
        charcmp(char,string)
        reverse(strlist,strlist,strlist)
goal
        clearwindow,
        start.
clauses
start:
-
        createlist(3,[],Newlist,List1), reverse(List1,[],List2),
        List2 = [Str1 |Tail], Tail = [Str2|Tail1],
        Tail1=[Str3|Str4],
        write("cmp string1 and string2"),nl, strcmp(Str1,Str2),
        write("cmp string1 and string3"),nl, strcmp(Str1,Str3),
        write("cmp string2 and string3"),nl, strcmp(Str2,Str3).
        createlist(Num,Oldlist,Newlist,List1):-
                Num > 0,
        write("Enter any string="), readln(Str),
        Newlist = [Str |Oldlist],
        NN = Num -1,
        createlist(NN,Newlist,List2,List1).
        createlist(_,Oldlist,_,List1):-
                List1 = Oldlist.
        strcmp(Str1,Str2):-
                frontchar(Str1,Ch1,Rest1), charcmp(Ch1,Str2),
        Rest1<>"",strcmp(Rest1,Str2).
strcmp(Str1,Str2).
charcmp(Ch1,Str2):-
frontchar(Str2,Ch2,Rest2),
Ch1<>Ch2,Rest2<>"",
charcmp(Ch1,Rest2).
charcmp(Ch1,Str2):-
        frontchar(Str2,Ch2,Rest2),
Ch1=Ch2,
write("char=",Ch2),nl.
charcmp(Ch1,Str2).
reverse([],Inputlist,Inputlist).
```

reverse([Head |Tail],List1,List2):-
reverse(Tail,[Head | List1],List2).

**OUTPUT :**



## PRACTICAL : 7 .

**Program (1):** Code for Prolog program for family hierarchy in Artificial Intelligence.

```
Predicates
        male(symbol).
        female(symbol).
        father(symbol,symbol).
        husband(symbol,symbol).
        brother(symbol,symbol).
        sister(symbol,symbol).
        listbrothers(symbol).
        listsisters(symbol).
        mother(symbol,symbol).
        grandfather(symbol).
        grandmother(symbol).
        uncle(symbol).
        aunt(symbol).
        cousin(symbol).
        listgrandsons(symbol).
        listgranddaughters(symbol).
        printmenu.
        action(integer).
        repeat.
clauses
        male(dashrath).
        male(ram).
        male(laxman).
        male(bharat).
        male(luv).
        male(kush).
        male(son_of_laxman).
        female(kaushalya).
        female(sita).
        female(urmila).
        female(daughter_of_dashrath).
        father(dashrath,ram).
        father(dashrath,laxman).
        father(dashrath,bharat).
        father(ram,luv).
father(ram,kush).
father(laxman,son_of_laxman).
father(dashrath,daughter_of_dashrath).
husband(dashrath,kaushalya).
husband(ram,sita).
```

```
husband(laxman,urmila).
mother(X,Y):-
        husband(Z,X),
        father(Z,Y).
brother(X,Y):-
        father(Z,X),
        father(Z,Y),
        X<>Y,
        male(X).
sister(X,Y):-
        father(Z,X),
        father(Z,Y),
        X<>Y,
        female(X).
listbrothers(X) :-
        brother(Z,X),
        write(Z).
listsisters(X):-
        sister(Z,X),
        write(Z).
grandfather(X):-
        father(Y,Z),
        father(Z,X),
        write(Y, " is the grandfather of ",X,"\n").
grandmother(X):-
        husband(Z,X),
        father(Z,V),
        father(V,Y),
        write(Y, " is the grandmother of ",X,"\n").
listgrandsons(X):-
        father(X,Z),
        father(Z,Y),
        male(Y),
        write(Y,"\n"),
        Fail.
listgrandsons(X):-
        husband(Y,X),
        father(Y,V),
        father(V,Z),
        male(Z),
        write(Z,"\n"), fail.
listgranddaughters(X):-
        father(X,Z),
        father(Z,Y),
```

```
        female(Y),
        write(Y,"\n"), fail.
listgranddaughters(X):-
        husband(Y,X), father(Y,V),
        father(V,Z),
        female(Z),
        write(Z,"\n"), fail.
uncle(X):-
        brother(Z,Y),
        father(Z,X),
        male(Y),
        write(Y,"\n"),
        fail.
aunt(X):-
        husband(Z,Y),
        brother(Z,V),
        father(V,X),
        write(Y,"\n"),
        fail.
cousin(X):-
        father(Z,X),
        father(V,Y),
        Z<>V,
        brother(V,Z),
        write(Y,"\n").
        repeat.
repeat:- repeat.
action(1):-
        write("\nEnter name of person whose father is to be found: "),
        readln(X),
        write("\n"),
        write("Father of ",X,"is:"),
        father(Z,X),
        write(Z,"\n"),
        fail.
action(2):-
        write("\nEnter name of person whose mother is to be found: "), readln(X),
        write("\n"),
        write("Mother of ",X," is:"),
        mother(Z,X), write(Z,"\n"), fail.
action(3):- write("\nEnter name of person whose brothers are to be found: "),
        readln(X),
        write("\n"),
        write("Brothers of ",X,"are:\n"),
```

```
        listbrothers(X),
        write("\n"),
        fail.
action(4):-
        write("\nEnter name of person whose sisters are to be found: "),
        readln(X),
        write("\n"),
        write("Sisters of ",X,"are:\n"),
        listsisters(X),
        write("\n"),
        fail.
action(5):-
        write("\nEnter name of person whose grandsons are to be found: "),
        readln(X),
        write("\n"),
        write("Grandsons of ",X,"are:\n"),
        listgrandsons(X),
        write("\n"),
        fail.
action(6):-
        write("\nEnter name of person whose granddaughters are to be found: "),
        readln(X),
        write("\n"),
        write("Granddaughters of ",X,"are:\n"),
        listgranddaughters(X),
        write("\n"),
        fail.
action(7):-
        write("\nEnter name of person whose uncles are to be found: "),
        readln(X),
        write("\n"),
        write("Uncles of ",X,"are:\n"),
        uncle(X),
        write("\n"),
        fail.
action(8):-
        write("\nEnter name of person whose aunties are to be found: "),
        readln(X),
        write("\n"),
        write("Aunties of ",X," are:\n"),
        aunt(X),
        write("\n"),
        fail.
```

```
action(9):-
        write("\nEnter name of person whose cousins are to be found: "),
        readln(X),
        write("\n"),
        write("Cousins of ",X,"are:\n"),
        cousin(X),
        write("\n"),
        fail.
action(0).printmenu:-
repeat,
write("\n1. Display Father of?\n"),
write("2. Display Mother of?\n"),
write("3. List all brothers of?\n"),
write("4. List all sisters of?\n"),
write("5. List all grandson of?\n"),
write("6. List all granddaughterof?\n"),
write("7. List all unclesof?\n"),
write("8. List all auntyof?\n"),
write("9. list all cousinsof?\n"),
write("0. exit\n"),
write("Enter your choice :"),
readInt(Choice),
action(Choice),
write("\n"), repeat.

goal
        makewindow(1,2,3,"Family
        Tree",0,0,25,80), printmenu.
```

**OUTPUT :**

```
+-----------------------------------------Family Tree-----------------------------------------+
|                                                                                             |
|1. Display Father of?                                                                        |
|2. Display Mother of?                                                                        |
|3. List all brothers of?                                                                     |
|4. List all sisters of?                                                                      |
|5. List all grandson of?                                                                     |
|6. List all granddaughter of?                                                                |
|7. List all uncles of?                                                                       |
|8. List all aunty of?                                                                        |
|9. list all cousins of?                                                                      |
|0. exit                                                                                      |
|Enter your choice : 1                                                                        |
|                                                                                             |
|Enter name of person whose father is to be found: ram                                        |
|                                                                                             |
|Father of ram is:dashrath                                                                    |
|                                                                                             |
|1. Display Father of?                                                                        |
|2. Display Mother of?                                                                        |
|3. List all brothers of?                                                                     |
|4. List all sisters of?                                                                      |
|5. List all grandson of?                                                                     |
|6. List all granddaughter of?                                                                |
|7. List all uncles of?                                                                       |
|8. List all aunty of?                                                                        |
|9. list all cousins of?                                                                      |
|0. exit                                                                                      |
|Enter your choice : 3                                                                        |
|                                                                                             |
|Enter name of person whose brothers are to be found: ram                                     |
|                                                                                             |
|Brothers of ram are:                                                                         |
|laxman                                                                                       |
|bharat                                                                                       |
|                                                                                             |
|1. Display Father of?                                                                        |
|2. Display Mother of?                                                                        |
|3. List all brothers of?                                                                     |
|4. List all sisters of?                                                                      |
|5. List all grandson of?                                                                     |
|6. List all granddaughter of?                                                                |
|7. List all uncles of?                                                                       |
|8. List all aunty of?                                                                        |
|9. list all cousins of?                                                                      |
```

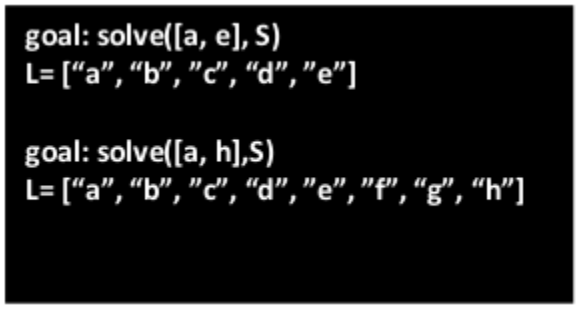
```
|0. exit                                                                    |
|Enter your choice : 5                                                      |
|                                                                           |
|Enter name of person whose grandsons are to be found: dashrath             |
|                                                                           |
|Grandsons of dashrath are:                                                 |
|luv                                                                        |
|kush                                                                       |
|son_of_laxman                                                              |
|                                                                           |
|1. Display Father of?                                                      |
|2. Display Mother of?                                                      |
|3. List all brothers of?                                                   |
|4. List all sisters of?                                                    |
|5. List all grandson of?                                                   |
|6. List all granddaughter of?                                              |
|7. List all uncles of?                                                     |
|8. List all aunty of?                                                      |
|9. list all cousins of?                                                    |
|0. exit                                                                    |
|Enter your choice : 7                                                      |
|                                                                           |
|Enter name of person whose uncles are to be found : kus                    |
|                                                                           |
|Uncles of kush are:                                                        |
|laxman                                                                     |
|bharat                                                                     |
|                                                                           |
|1. Display Father of?                                                      |
|2. Display Mother of?                                                      |
|3. List all brothers of?                                                   |
|4. List all sisters of?                                                    |
|5. List all grandson of?                                                   |
|6. List all granddaughter of?                                              |
|7. List all uncles of?                                                     |
|8. List all aunty of?                                                      |
|9. list all cousins of?                                                    |
|0. exit                                                                    |
|Enter your choice :                                                        |
|                                                                           |
|                                                                           |
|Press the SPACE bar                                                        |
+---------------------------------------------------------------------------
```

## PRACTICAL : 8

**Program** : Write a program to implement BFS (for AI search problem).

```
domains
        X, H, N, ND=symbol
        P, L, T, Z, Z1, L1, L2, L3, PS, NP, ST, SOL=symbol*
Predicates
        solve(L, L) member(X,L) extend(L, L) conc(X, L, L) breadthfirst(L, L) goal(X)
clauses
        solve(start, solution):-/*solution is a state from start to a goal*/ breadthfirst ([[start]],
        solution).
        breadthfirst([[node|path]| _ ] ,[node|path]):- /*solution is an extension to a goal*/
        /*of one of path*/
goal(node).
breadthfirst([path|paths], solution):- extend(path,newpaths), conc(paths,newpaths,path1),
breadthfirst(path1,solution).
extend([node|path],newpaths):- bagof([newnode, node|path],(s(node,
newnode),notmember(newnode,[node|path])), newpaths),!. extend(path, []).
conc([], L, L).
conc([X|L1], L2, [X|L3]):- conc(L1, L2, L3).
member(X, [X|T]).
member(X,
[H|T]):-
member(X, T).
```

**OUTPUT :**



## PRACTICAL : 9

**Program** : Write a program to implement DFS (for 8 puzzle problem)

```
domains
H=integer T=integer*
predicates
        safe(T)
        solution(T)
        permutation(T,T)
        del(H,T,T)
        noattack(H,T,H)
clauses
        del(I,[I|L],L). /*to take a position from the permutation oflist*/
        del(I,[F|L],[F|L1]):-
        del(I,L,L1).
        permutation([],[]).       /*to find the possiblepositions*/
        permutation([H|T],PL):-
                permutation(T,PT),\ del(H,PL,PT).
        solution(Q):- /*final solution is stored inQ*/
                permutation([1,2,3,4,5,6,7,8],Q),
                safe(Q).
                safe([]). /*Q is safe such that no queens attack eachother*/
                safe([Q|others]):-
                safe(others),
                noattack(Q,others,1).
                noattack(_,[],_)          ./*to find if the queens are in same row, column or
        diagonal*/
                 noattack(Y,[Y1|Ydist],Xdist):-
Y1-Y>Xdist,
Y-Y1<>Xdist,
dist1=Xdist,
noattack(Y,Ydist,dist1).
```

## OUTPUT:-

```
goal:-solution(Q). Q=["3","8","4","7","1","6",2","5"]
```

## PRACTICAL : 10

**Program :** Write a program to implement A* Algorithm.

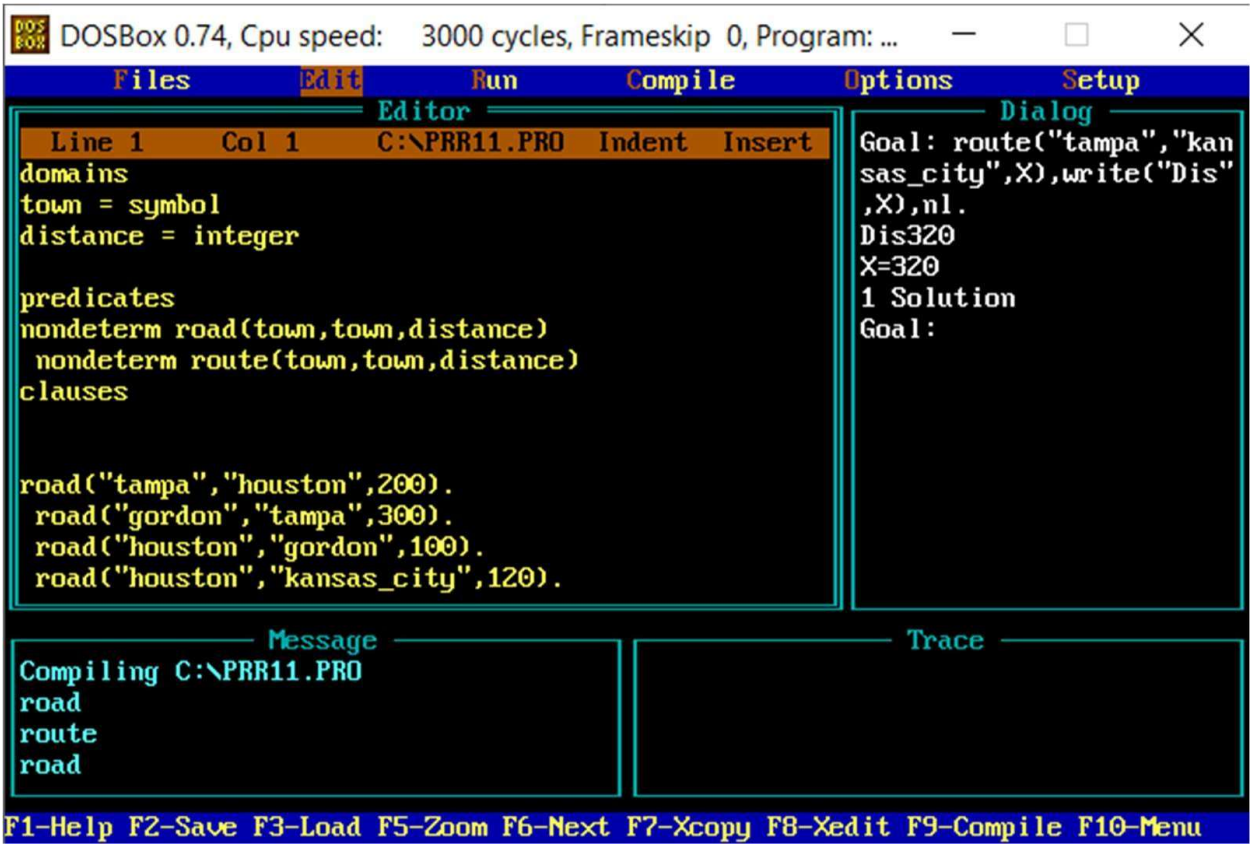
```
%%%
%%%
%%% Nodes have form S#D#F#A
%%% where S describes the state orconfiguration
%%%D is the depth of the node
%%%F is the evaluation function value
%%%A is the ancestor list for the node
:- op(400,yfx,'#'). /* Node builder notation*/
solve(State,Soln) :- f_function(State,0,F),
search([State#0#F#[]],S), reverse(S,Soln).
f_function(State,D,F) :- h_function(State,H),
F is D + H.
search([State#_#_#Soln|_], Soln) :-
goal(State). search([B|R],S) :-
expand(B,Children),
insert_all(Children,R,Ope
n), search(Open,S).
insert_all([F|R],Open1,Open3) :- insert
(F,Open1,Open2),
insert_all(R,Open2,Open3).
insert_all([],Open,Open).
insert(B,Open,Open) :- repeat_node(B,Open), !
. insert(B,[C|R],[B,C|R]) :- cheaper(B,C), ! .
insert(B,[B1|R],[B1|S]) :- insert(B,R,S),
!. insert(B,[],[B]).
repeat_node(P#_#_#_, [P#_#_#_|_]).
cheaper( _#_#F1#_ , _#_#F2#_ ) :- F1 <
F2.
expand(State#D#_#S,All_My_Childr
en) :-
bagof(Child#D1#F#[Move|S],
(D1 is D+1,
move(State,Child,Mov
e),
f_function(Child,D1,F))
, All_My_Children).
```

## PRACTICAL : 11

**Program** : Write a program to solve travelling salesman problem using Prolog.

```
domains
        town = symbol
        distance = integer
predicates
        Nondeterm road(town,town,distance)
        nondeterm route(town,town,distance)
clauses
        road("tampa","houston",200).
        road("gordon","tampa",300).
        road("houston","gordon",100).
        road("houston","kansas_city",120).
        road("gordon","kansas_city",130).
        route(Town1,Town2,Distance):-
        road(Town1,Town2,Distance).
        route(Town1,Town2,Distance):-
        road(Town1,X,Dist1),
        route(X,Town2,Dist2),
        Distance=Dist1+Dist2, !.
```

**OUTPUT :**



## PRACTICAL : 12

**Program** : Study of dynamic database in PROLOG.

```
Predicates
        reading
        writing
        delete
        find(integer)
        startup(integer)
Database
        unsortedDatabase(string,integer)
        sortedDatabase(string)
Clauses
        startup(0).
        startup(Num):-
        write("Enter String = "),
        readln(Name),
        str_len(Name,Len),
        asserta(unsortedDatabase(Name,Len)),
        TempNum = Num - 1,
        startup(TempNum).
        writing:-
        sortedDatabase(Name),
        write(Name),nl,
        fail.
        writing.
        find(Index):-
        unsortedDatabase(Name,Index),
        assertz(sortedDatabase(Name)),
        retract(unsortedDatabase(Name,Index)),
        find(Index).
        find(Index):-
        Index = 255.
        find(Index):-
        TempIndex = Index + 1,
        find(TempIndex).
        reading:-
        NumRead = 10,
        startup(NumRead).
        delete :-
        retract(sortedDatabase(_)),
        fail.
        delete.
Goal
        Clearwindow,
```

```
makewindow(1,2,3,"String Operations",0,0,25,80),
reading,!,
find(1),
write("\nString In Increasing Order Of Their Length Are : \n"),
writing,
Delete.
```

**OUTPUT :**