

**Hindi Vidya Prachar Samiti's**  
**RAMNIRANJAN JHUNJHUNWALA COLLEGE OF**  
**ARTS, SCIENCE & COMMERCE**  
**(AUTONOMUS)**

**Artificial Intelligence**



**Name:** Richard Caleb

**Roll No.:** 735

**Class:** MSc Data Science and Artificial Intelligence Part-I



## **Ramniranjan Jhunjhunwala College of Arts, Science and Commerce**

### **Department of Data Science and Artificial Intelligence**

# **CERTIFICATE**

This is to certify Mast. Richard Caleb of MSc. Data Science and Artificial Intelligence, Roll no. 735 has successfully completed the practical of ARTIFICIAL INTELLIGENCE during the Academic Year 2023-2024.

Date:

Prof. Sujata Kotian  
(Prof-In-Charge)

External Examiner

INDEX			
Sr. No	Practical Name	Date	Signature
1.	Supervised learning using Weka tool a) Decision Tree b) Logistic Regression c) Support Vector Machine d) Linear Search	09/10/2023	
2.	Unsupervised learning using Weka tool a) Clustering b) Hierarchical Clustering c) Density based Clustering d) Simple K-Means Clustering	09/10/2023	
3.	a) Association Algorithm b) Naïve Bayes Classifier c) Feature extraction	10/10/2023	
4.	Prolog Problem a) to implement simple query and facts b) query and facts on Foods c) query and facts on Student teacher relationship d) to implement simple query and facts on Cars e) query and facts on Cat Relationship	13/10/2023	
5.	Prolog Problem a) to find max and min of 2 nos. b) arithmetic expression	14/10/2023	
6.	Prolog Problem a) calculate sum of 2 nos. using variable b) calculate sum of 2 nos. and store in 3 <sup>rd</sup> variable c) to implement max(X,Y,M) d) to implement multiplication	17/10/2023	
7.	Prolog Problem a) to find cube number b) implement addition and multiplication taking input from user	17/10/2023	
8.	To implement Knowledge Base	17/10/2023	
9.	Tower Hanoi	17/10/2023	

## GENERAL TERMINOLOGY

**Kappa's Statistics:** Measures the precision of data items.

Used to determine the chance of agreement due to guessing a possibility in the same way the chances of correct answers possible on multiple test.

**Absolute Error:** Amount of Error calculated

**Mean Absolute Error:** Average of all absolute error

**Root Mean Squared Error:** It measures the difference between the values which are predicted by a model and the actual values.

**Root Absolute Error:** The Absolute Error gives how large the error is while the Relative Error gives how large the error is related to correct value.

**Root Relative Squared Error:** It is a relative to what it would have been if a simple predictor has been used. tp: True Positive fp: False Positive tn: True Negative fn: False Negative

**Precision:** Almost near to accuracy

**Recall:** Proportion of instances classified as a given class / Actual total in that class

**f-measure:** Combine measure for precision and recall

**MCC (Matthew's Correlation co-efficient):** Measure the quality of binary classification. It takes into account of true and false

**ROC:** Receiver Operating Characteristics. It gives an idea how classifier are performing.

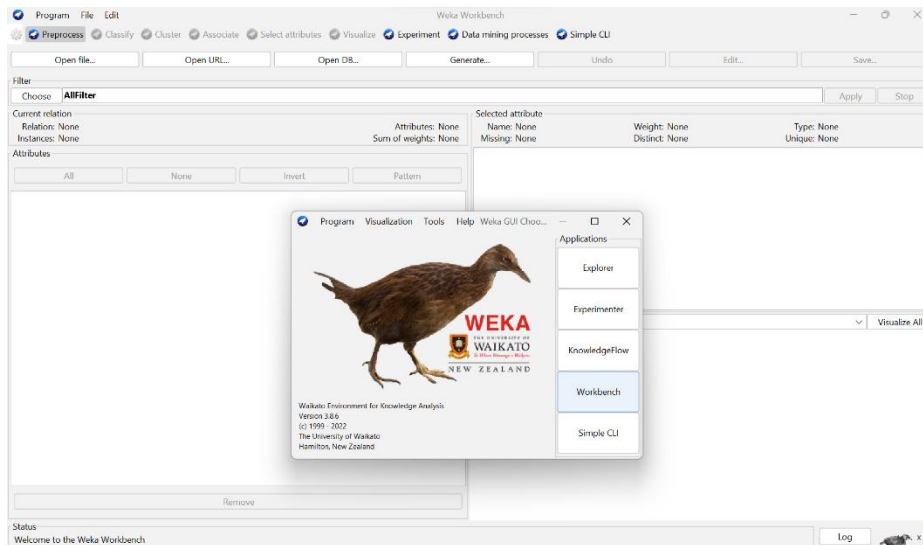
# PRACTICAL – I

Date: 09/10/2023

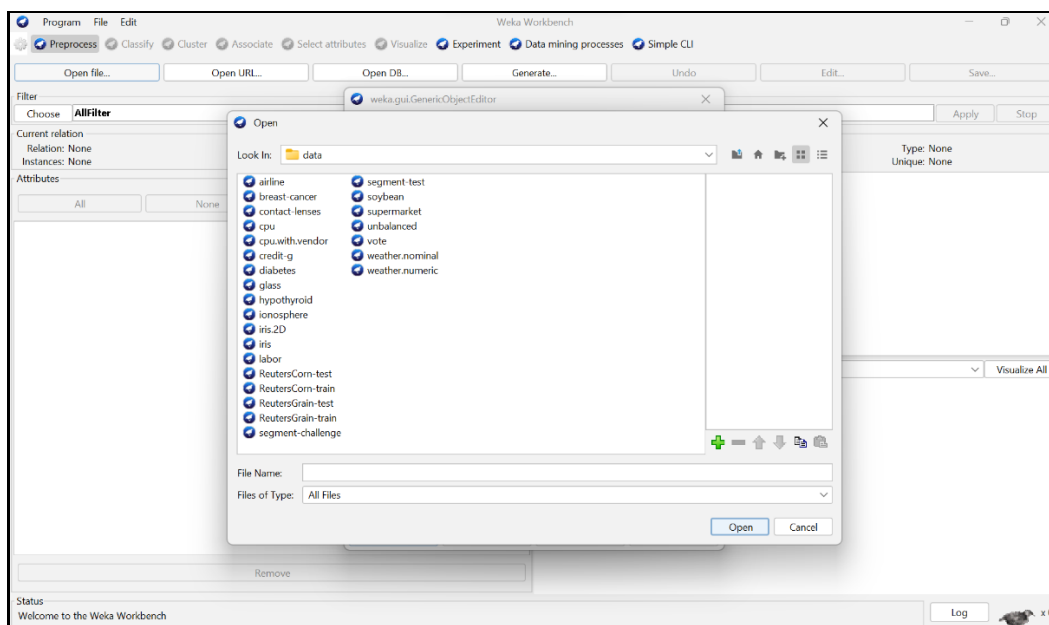
## AIM: Supervised Learning using Weka Tool (Decision Tree)

### STEPS TO BE FOLLOWED:

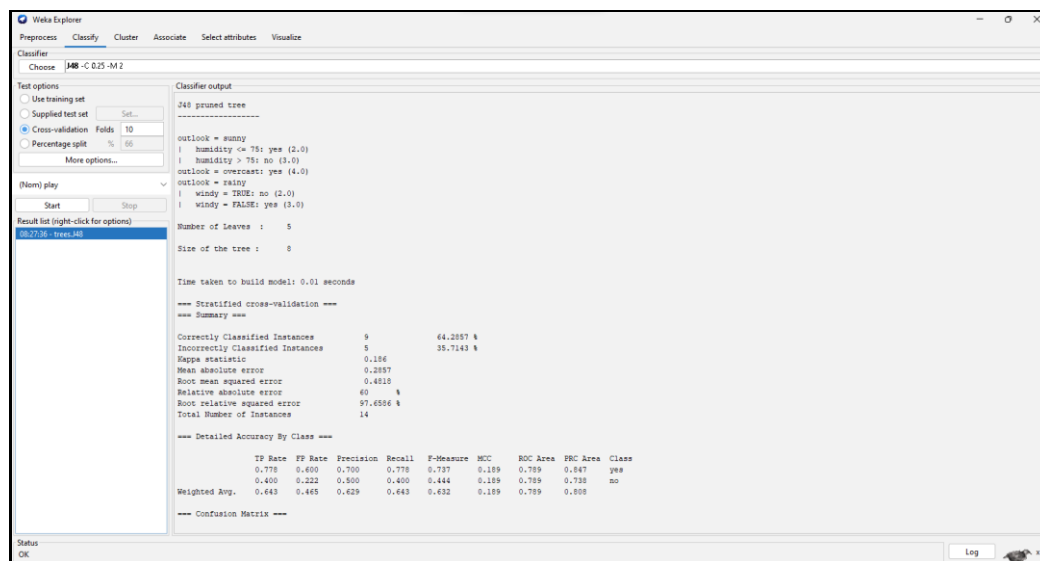
#### STEP 1: Install and Open Weka in Workbench Mode



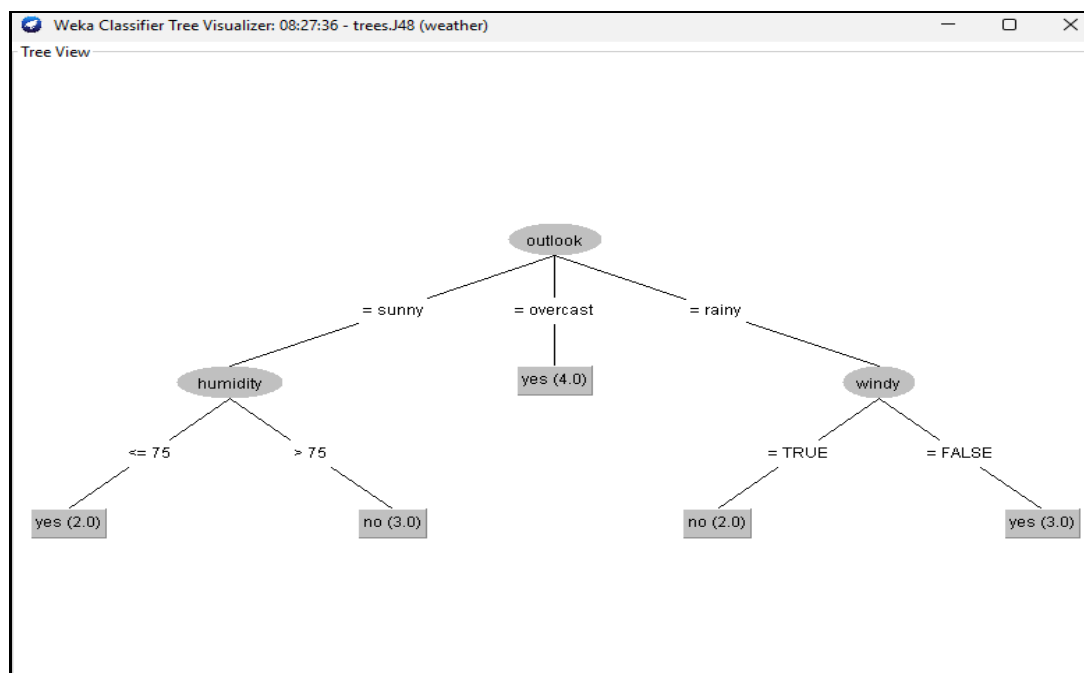
STEP 2: Click on the option “Open file..” which is right below Preprocess >> In that go to C:\ Program files\ Weka through “Look in” >> Under Weka go to data to upload the desired dataset for performing the given algorithm. (We are using Weather dataset here)



STEP 3: Go to option Classify >> Choose (Classifier) >> Tress >> J48 (for performing decision tree) >> click on Start



STEP 4: To Visualize the decision tree>> right click on tree.J48 >> Click on Visualize tree.



# PRACTICAL – I (b)

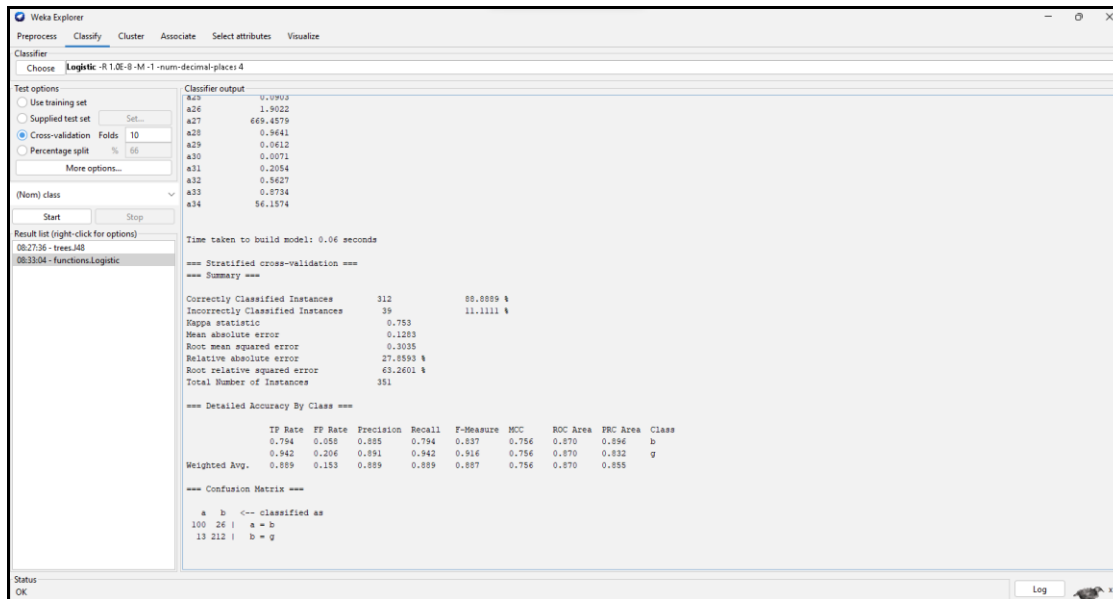
Date: 09/10/2023

## AIM: Supervised Learning using Weka Tool (Logistic Regression)

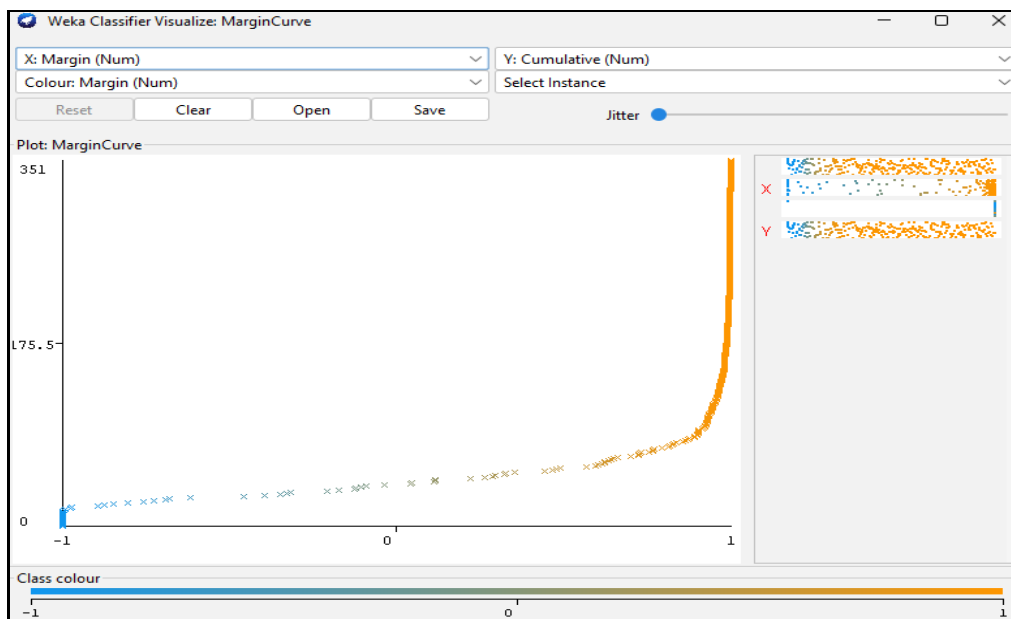
### STEPS TO BE FOLLOWED:

Follow the same steps until step 2, Here instead of weather dataset we will be using dataset name ionosphere.

STEP 3: Go to option Classify >> Choose (Classifier) >> Functions >> Logistic >> click on Start



STEP 4: To Visualize >> function.Logistic >> Click on Visualize MarginCurve



# PRACTICAL – I (c )

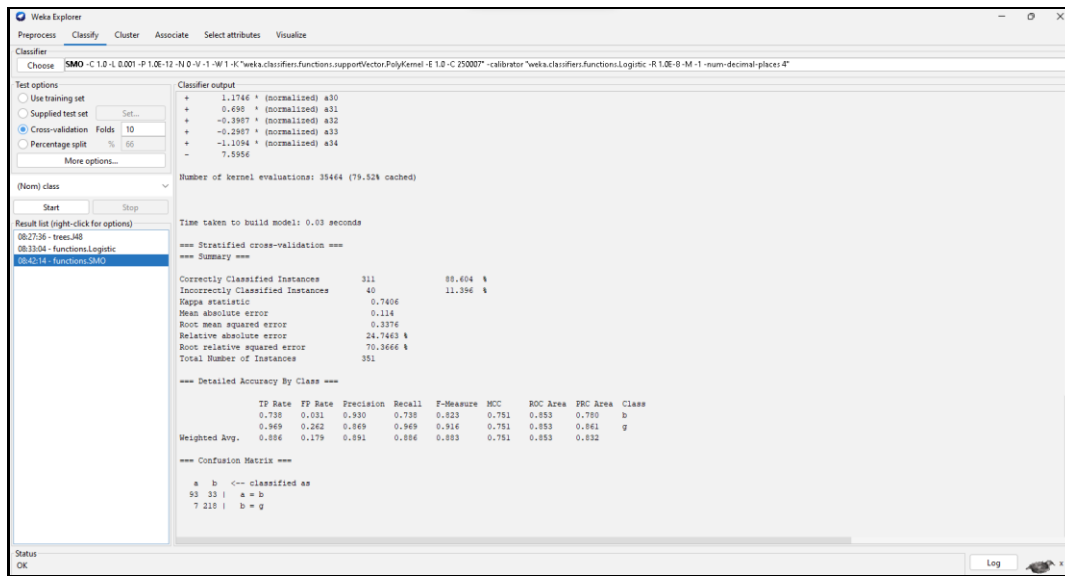
Date: 09/10/2023

## AIM: Supervised Learning using Weka Tool (Support Vector Machine)

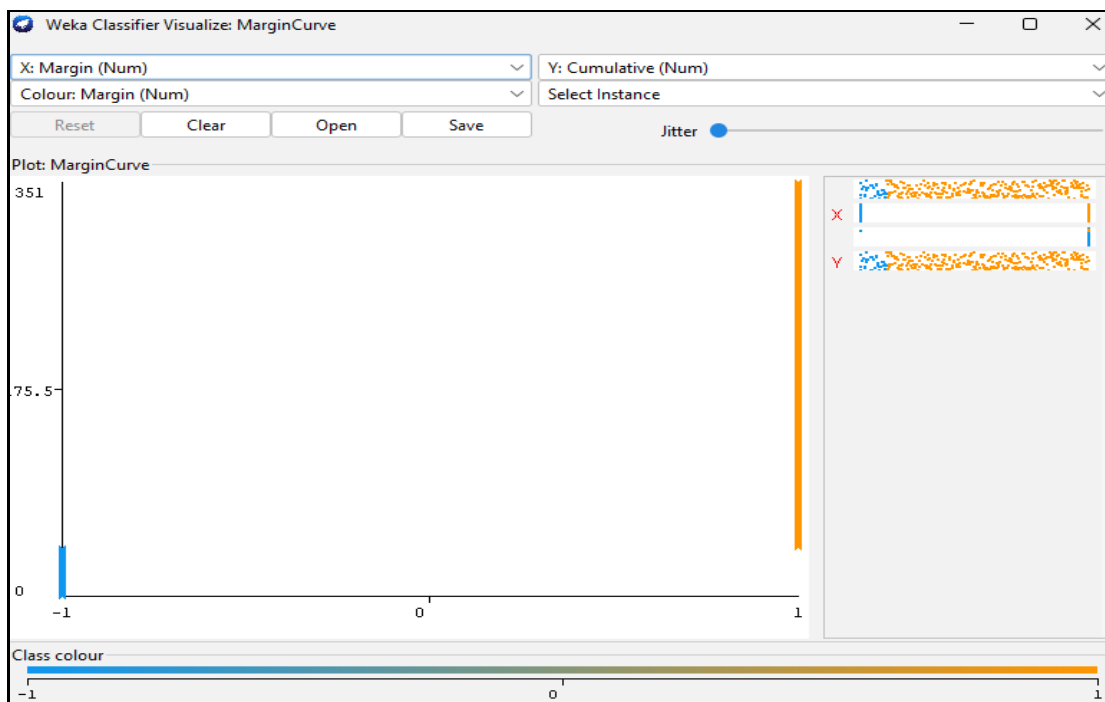
### STEPS TO BE FOLLOWED:

Follow the same steps until step 2, Here instead of weather dataset we will be using dataset name ionosphere.

STEP 3: Go to option Classify >> Choose (Classifier) >> Functions >> SMO>> click on Start



STEP 4: To Visualize >> function.SMO >> Click on Visualize MarginCurve





# PRACTICAL – I (d)

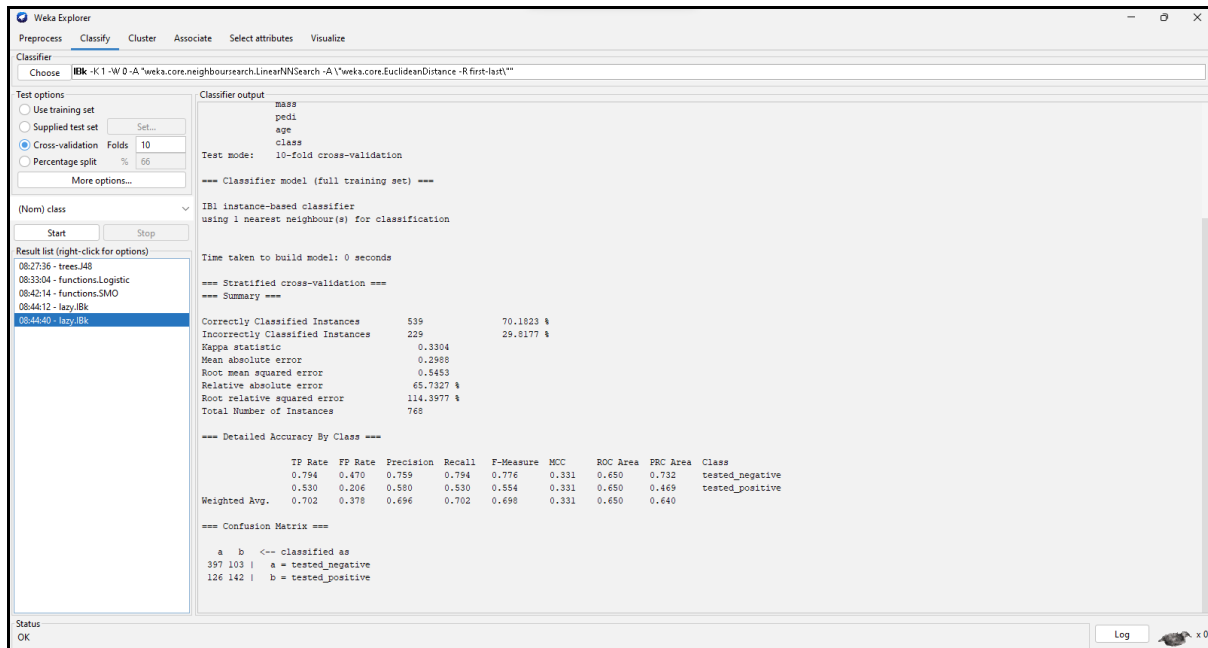
Date: 09/10/2023

## AIM: Supervised Learning using Weka Tool (Linear Search)

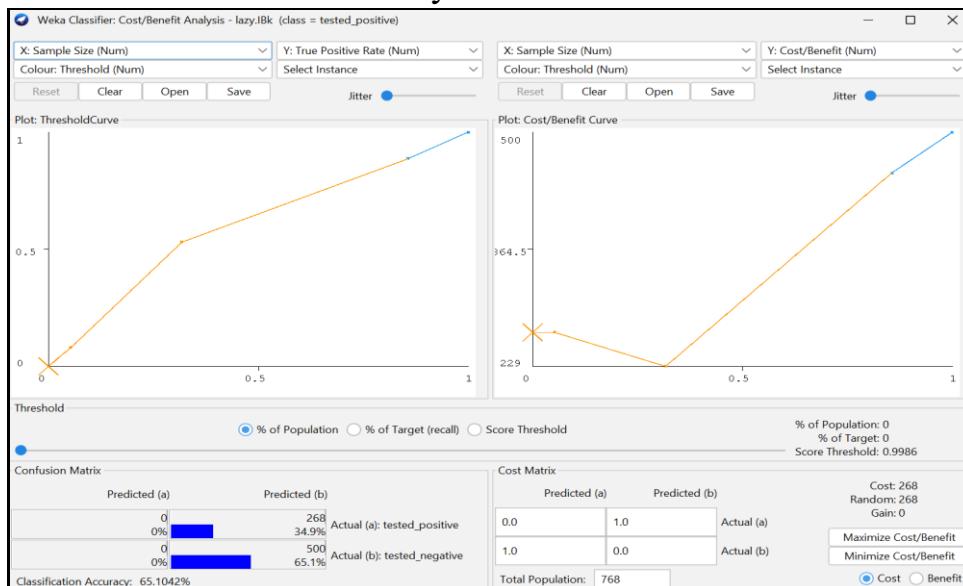
### STEPS TO BE FOLLOWED:

Follow the same steps until step 2, Here instead of weather dataset we will be using dataset name diabetes.

STEP 3: Go to option Classify >> Choose (Classifier) >> Choose >> Lazy >> IBk >> click on Start



STEP 4: To Visualize >> lazy.IBk >> Click on Cost/Benefit Analysis



SIGNATURE: \_\_\_\_\_

# PRACTICAL – II (a)

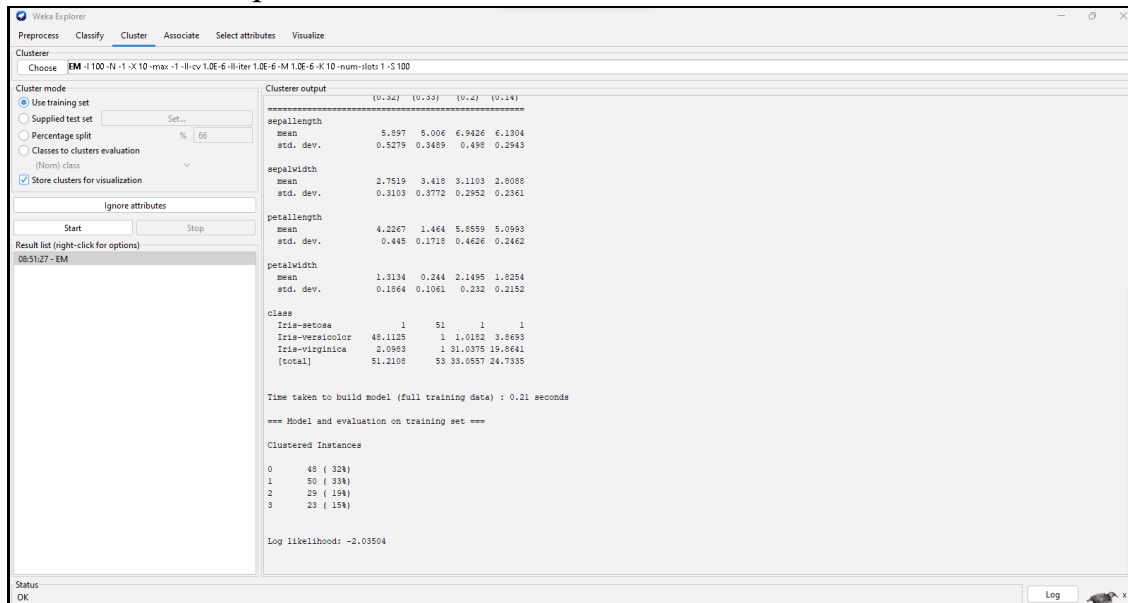
Date: 09/10/2023

## AIM: Unsupervised Learning using Weka Tool (Clustering)

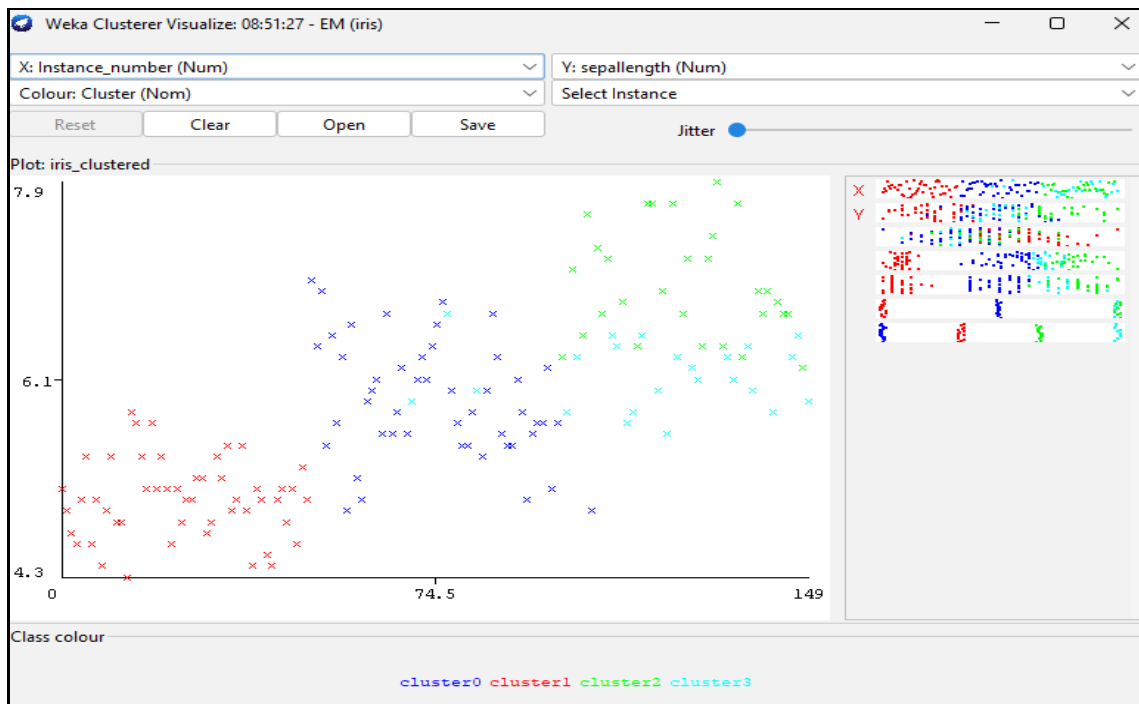
### STEPS TO BE FOLLOWED:

Follow the same steps until step 2, Here instead of weather dataset of prac1 we will be using dataset name iris.

STEP 3: Go to option Cluster >> Choose >> EM >> click on Start.



STEP 4: To Visualize >> EM >> Click on Visualize



## PRACTICAL – II (b)

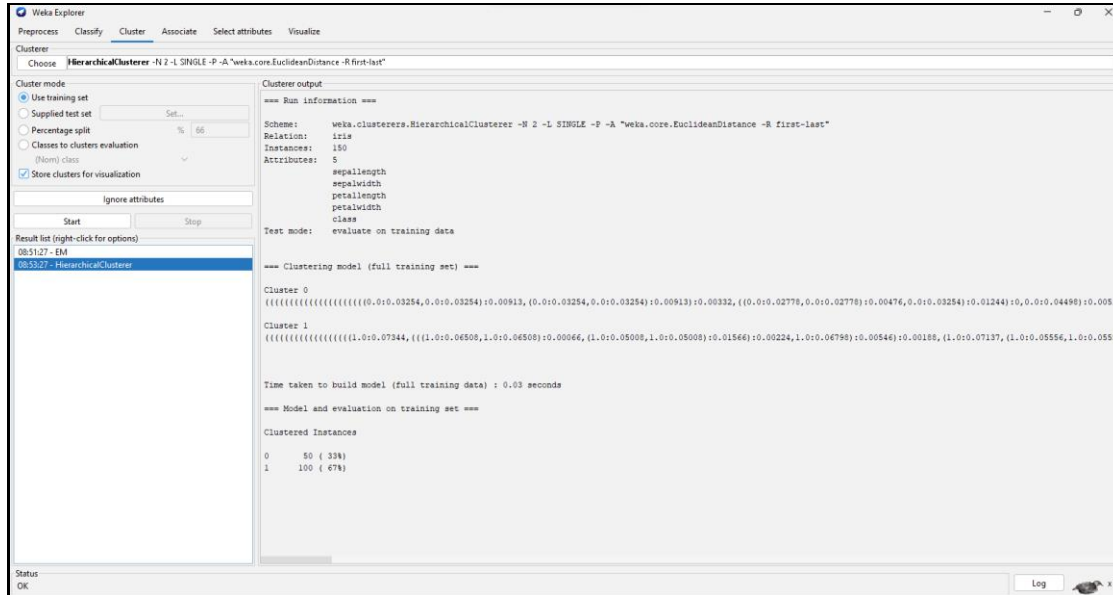
Date: 09/10/2023

### AIM: Unsupervised Learning using Weka Tool (Heirarchical Clustering)

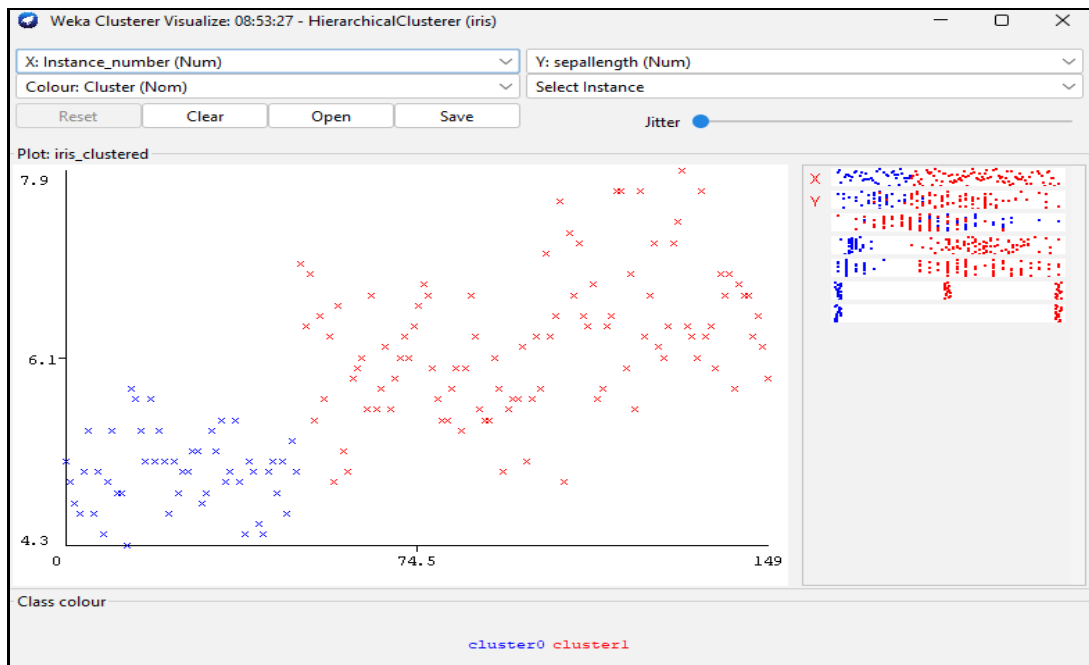
#### STEPS TO BE FOLLOWED:

Follow the same steps until step 2, Here instead of weather dataset of prac1 we will be using dataset name iris.

STEP 3: Go to option Cluster >> Choose >> Heirarchical>> click on Start.



STEP 4: To Visualize >>Heirarchical >> Click on Visualize



## PRACTICAL – II (c )

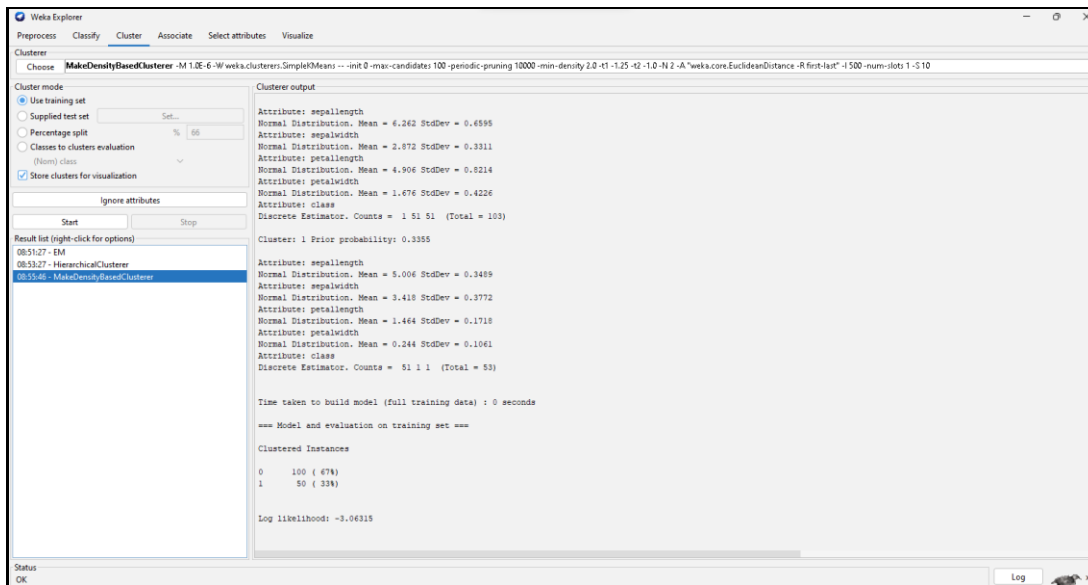
Date: 09/10/2023

### AIM: Unsupervised Learning using Weka Tool (Density Based Clustering)

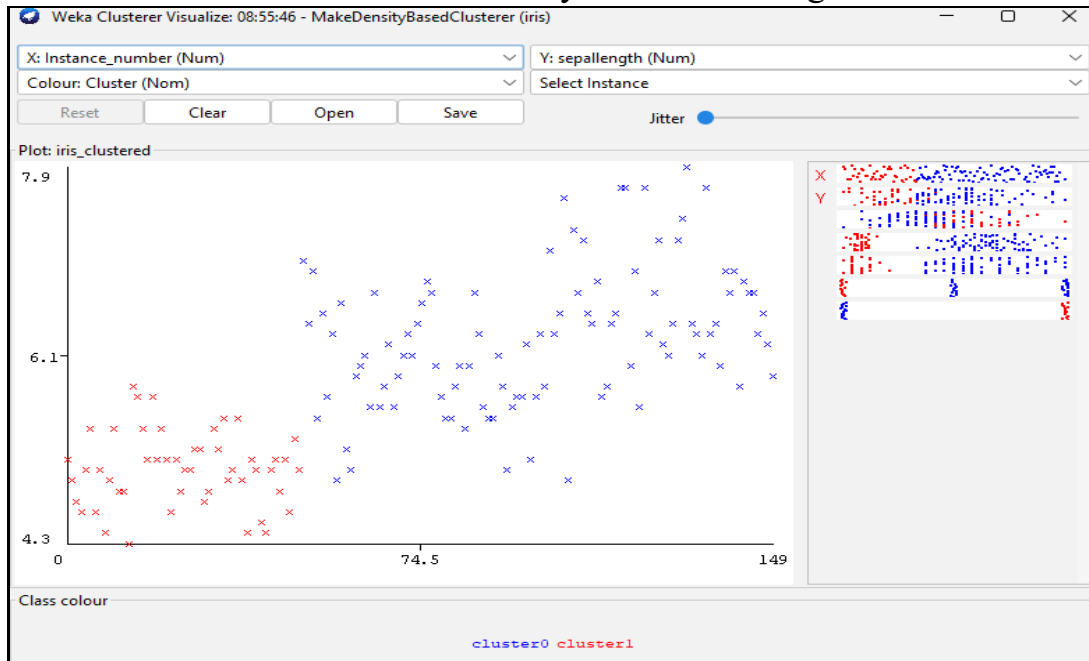
#### STEPS TO BE FOLLOWED:

Follow the same steps until step 2, Here instead of weather dataset of prac1 we will be using dataset name iris.

STEP 3: Go to option Cluster >> Choose >> MakeDensityBasedClustering >> click on Start.



STEP 4: To Visualize >> MakeDensityBasedClustering >> Click on Visualize



## PRACTICAL – II (d)

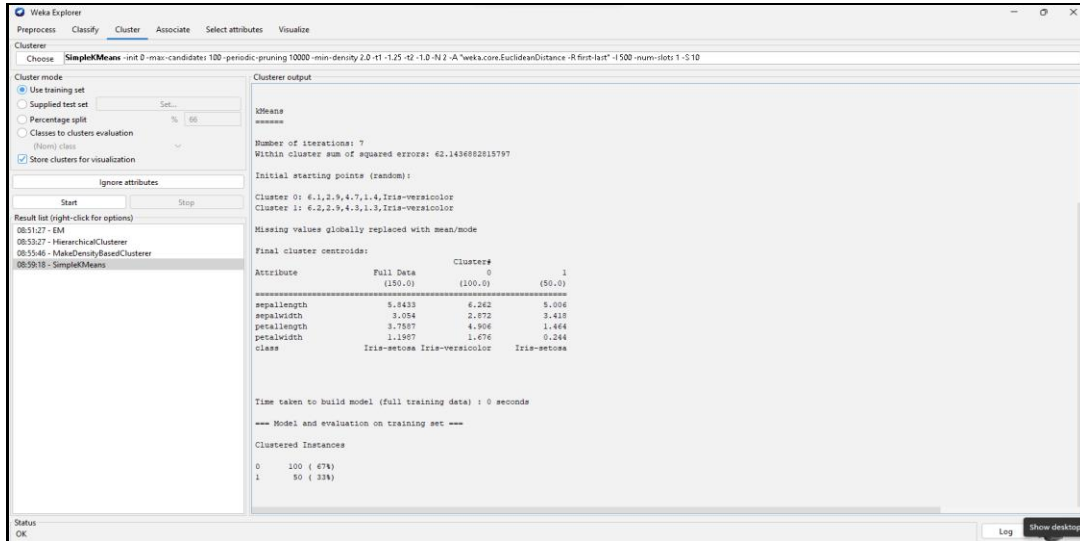
Date: 09/10/2023

### AIM: Unsupervised Learning using Weka Tool (Simple KMeans)

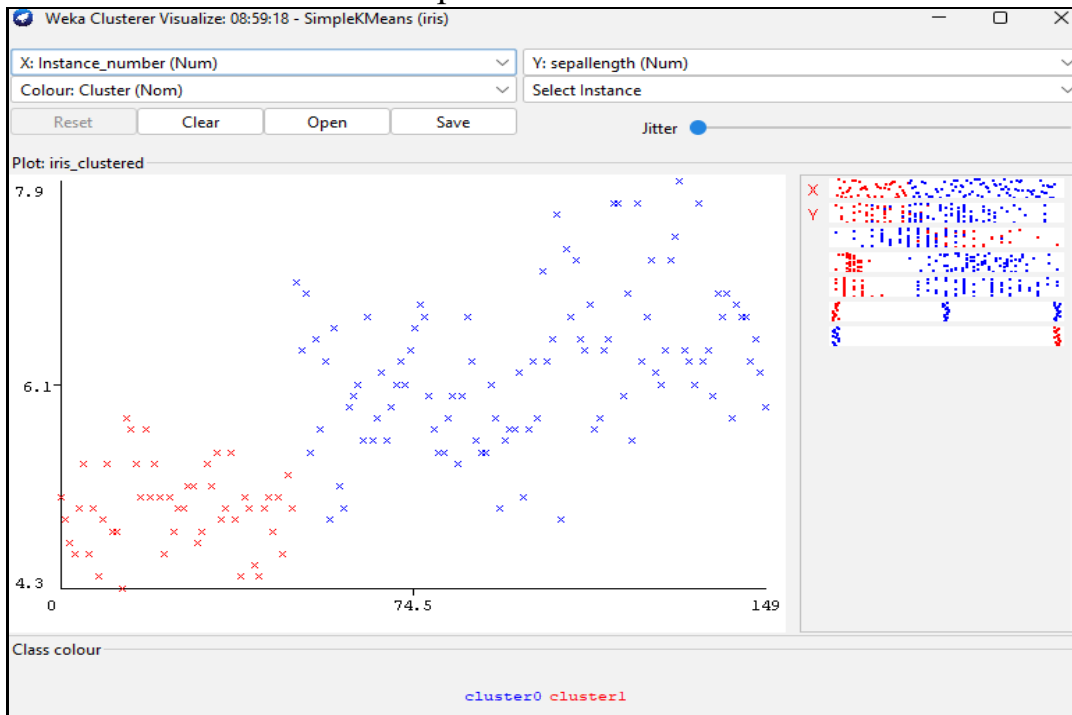
#### STEPS TO BE FOLLOWED:

Follow the same steps until step 2, Here instead of weather dataset of prac1 we will be using dataset name iris.

STEP 3: Go to option Cluster >> Choose >> Simple KMeans >> click on Start.



STEP 4: To Visualize >> SimpleKMeans >> Click on Visualize



SIGNATURE: \_\_\_\_\_

## PRACTICAL – III (a)

Date: 10/10/2023

### AIM: Association algorithm (Apriori Algo) using Weka Tool

#### STEPS TO BE FOLLOWED:

Follow the same steps until step 2 of prac1, Here instead of weather dataset of prac1 we will be using dataset name supermarket.

STEP 3: Go to option Associate >> Choose >> Apriori >> click on Start.



## PRACTICAL – III (b)

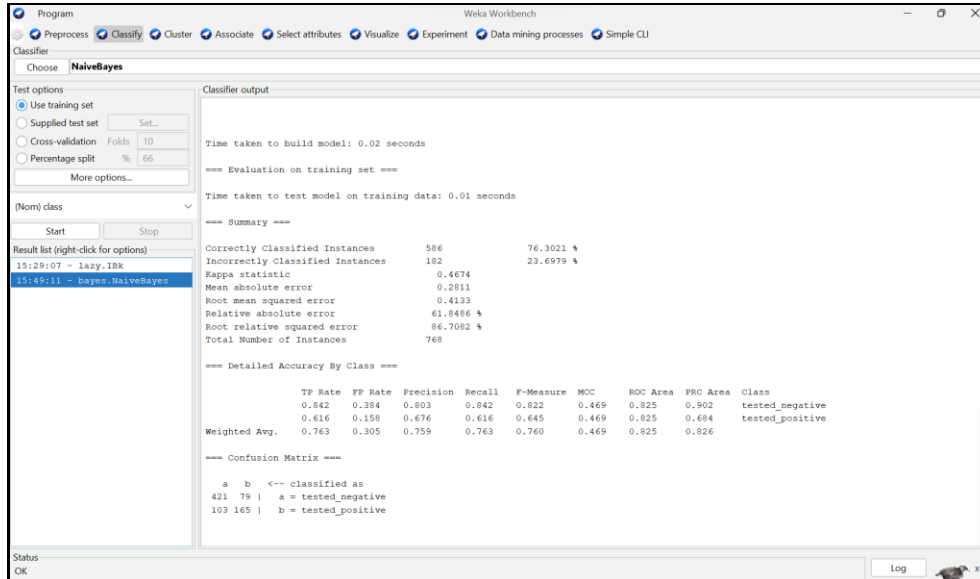
Date: 10/10/2023

### AIM: Naïve Bayes using Weka Tool

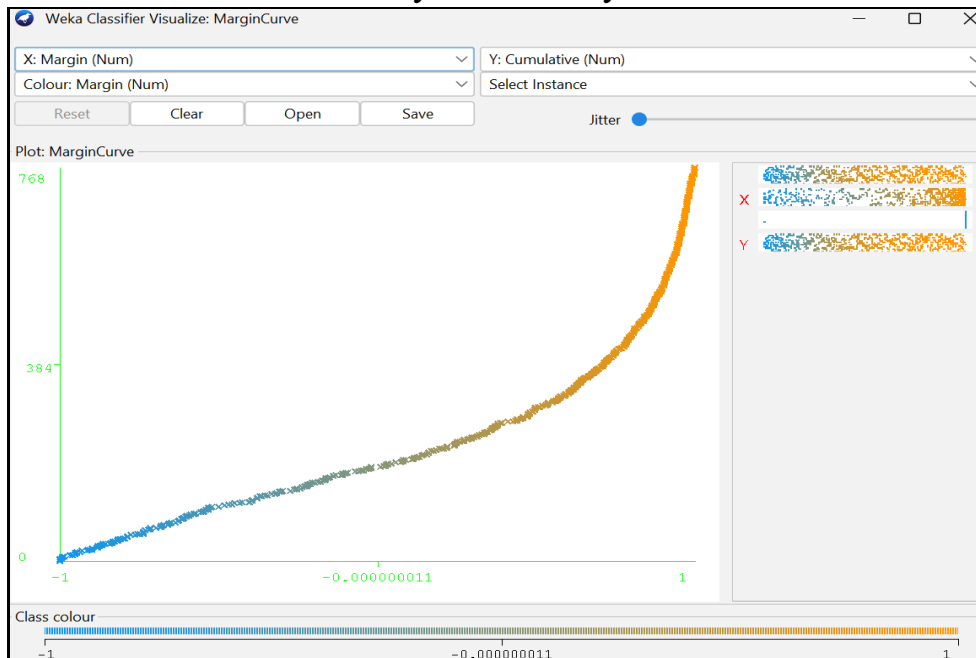
#### STEPS TO BE FOLLOWED:

Follow the same steps until step 2 of prac1, Here instead of weather dataset of prac1 we will be using dataset name diabetes.

STEP 3: Go to option Cluster >> Choose >> Bayes >> Naïve Bayes >> click on Start



STEP 4: To Visualize >> bayes.NaiveBayes >> Click on Visualize for Margin Curve



# PRACTICAL – III (c)

Date: 10/10/2023

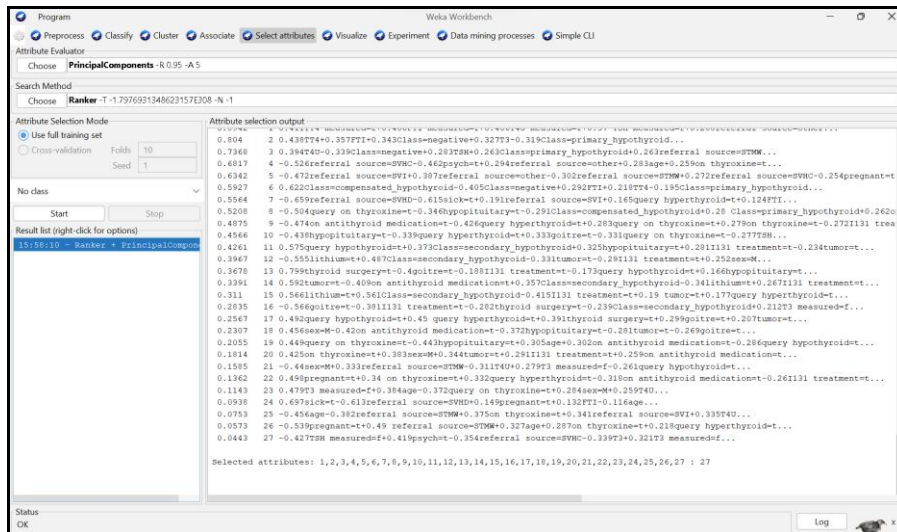
## AIM: Feature Extraction using Weka Tool

### STEPS TO BE FOLLOWED:

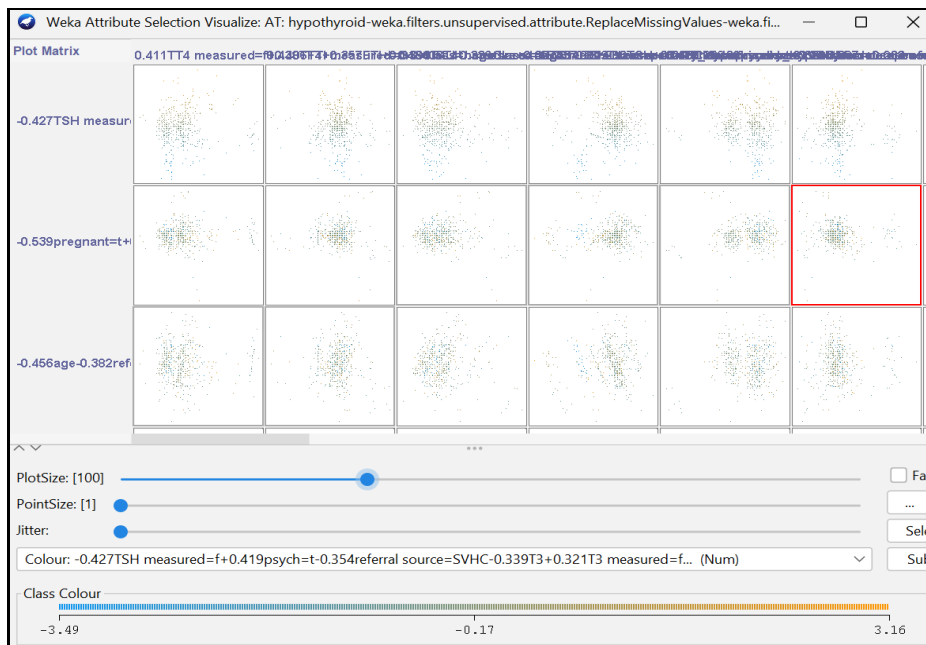
Follow the same steps until step 2 of prac1, Here instead of weather dataset of prac1 we will be using dataset name hypothyroid.

STEP 3: Go to option Selectattributes>> Choose (Attribute Evaluator)

>>PrincipalComponent>>Search Method (Choose) >> Ranker (which feasible with PCA)



STEP 4: To Visualize >>Ranker + PCA >> Click on Visualize transformed data



SIGNATURE: \_\_\_\_\_



## PRACTICAL – IV(a)

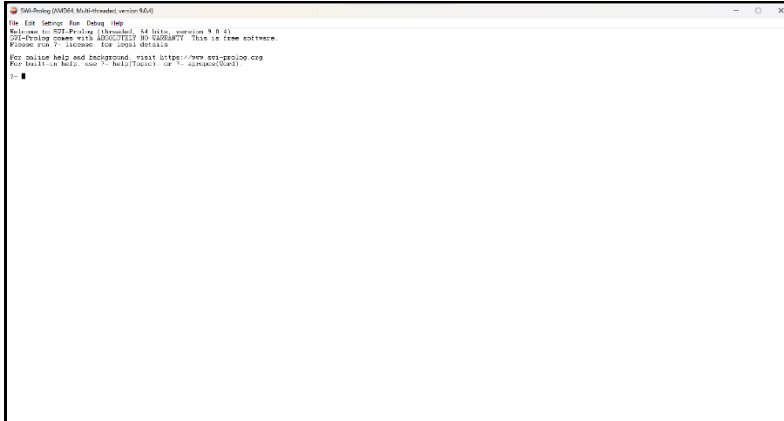
Date: 13/10/2023

## AIM: Basic Prolog Problem

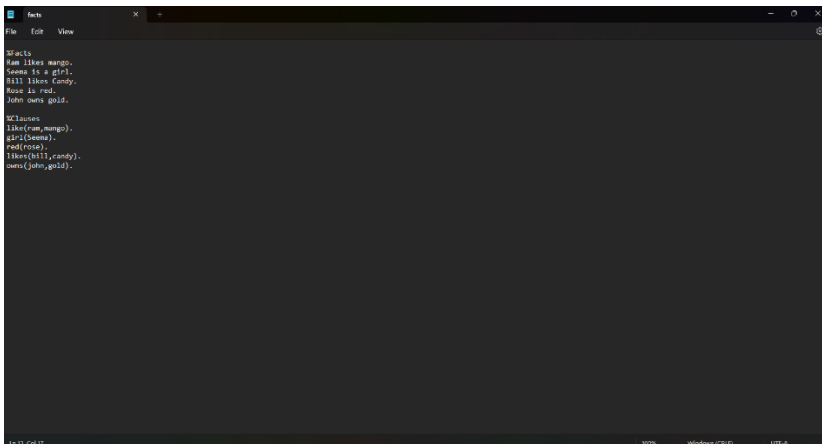
**Write a program in prolog to implement simple facts & Queries.**

### STEPS TO BE FOLLOWED:

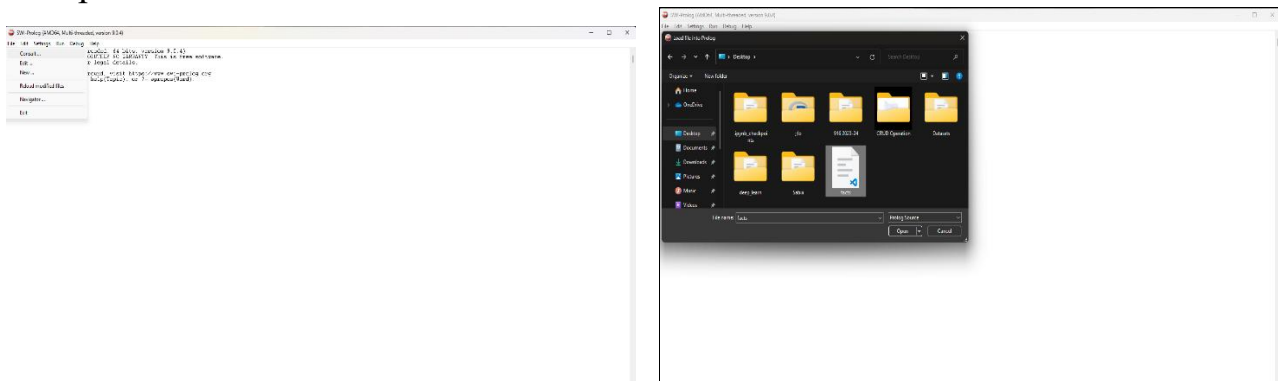
## STEP 1: Download and Open SWI prolog



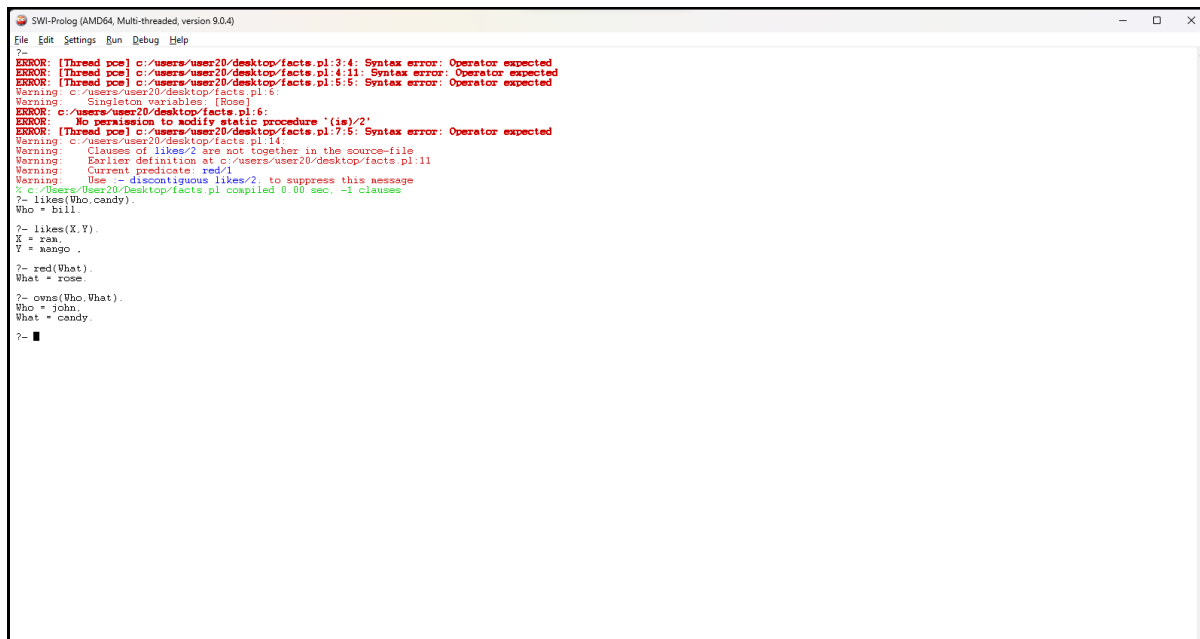
STEP 2: Open the Notepad and Write down the desired facts and clauses and save it with the “.pl” extension.



STEP 3: Go to File option at the console >> Consult >> and upload the data saved in the notepad.



STEP 4: After successfully uploading the data into SWI-Prolog and getting the message as clauses compiled, write the desired queries.



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
?-
ERROR: [Thread pce] c:/users/user20/desktop/facts.pl:3:4: Syntax error: Operator expected
ERROR: [Thread pce] c:/users/user20/desktop/facts.pl:4:11: Syntax error: Operator expected
ERROR: [Thread pce] c:/users/user20/desktop/facts.pl:5:8: Syntax error: Operator expected
Warning: c:/users/user20/desktop/facts.pl:6:
Warning: Singleton variables: [Rose]
ERROR: c:/users/user20/desktop/facts.pl:6:
ERROR: No permission to modify static procedure '(is)/2'
ERROR: [Thread pce] c:/users/user20/desktop/facts.pl:7:5: Syntax error: Operator expected
Warning: c:/users/user20/desktop/facts.pl:14:
Warning: Clauses of likes/2 are not together in the source-file
Warning: Earlier definition at c:/users/user20/desktop/facts.pl:11
Warning: Current predicate: red/1
Warning: Use :- disjointlikes/2. to suppress this message
% c:/Users/User20/Desktop/facts.pl compiled 0.00 sec, -1 clauses
?- likes(Who,candy).
Who = bill.

?- likes(X,Y).
X = ran,
Y = mango.

?- red(What).
What = rose.

?- owns(Who,What).
Who = john,
What = candy.

?-
```

## PRACTICAL – IV(b)

Date: 13/10/2023

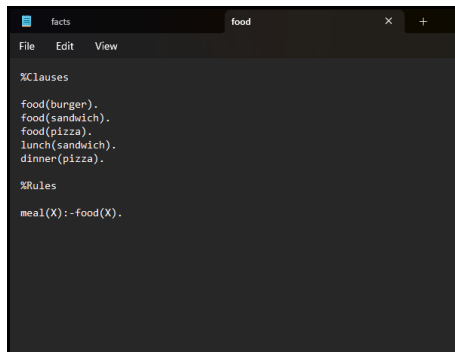
### AIM: Basic Prolog Problem

Write a program in prolog to implement simple facts & Queries (foods)

### STEPS TO BE FOLLOWED:

Follow the same steps as above, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.



```
File Edit View

%Clauses
food(burger).
food(sandwich).
food(pizza).
lunch(sandwich).
dinner(pizza).

%Rules
meal(X) :- food(X).
```

```
?-
ERROR: [Thread pce] c:/users/user20/desktop/food.pl:11:17: Syntax error: Unexpected end of file
% c:/Users/User20/Desktop/food.pl compiled 0.00 sec, 5 clauses
% c:/Users/User20/Desktop/food.pl compiled 0.00 sec, 1 clauses
?-
|   food(pizza).
true.
?- meal(X),lunch(X).
X = sandwich
Unknown action: i (h for help)
Action?
Unknown action: k (h for help)
Action?
ERROR: Type error: 'character_code' expected, found '-1' (an integer)
ERROR: In:
ERROR: [11] char_code(_1914,-1)
ERROR: [10] '$in_reply'(-1,'?h') at c:/program files/swipl/boot/init.pl:1037
?- dinner(sandwich).
false.
?-
```

## PRACTICAL – IV(c)

Date: 13/10/2023

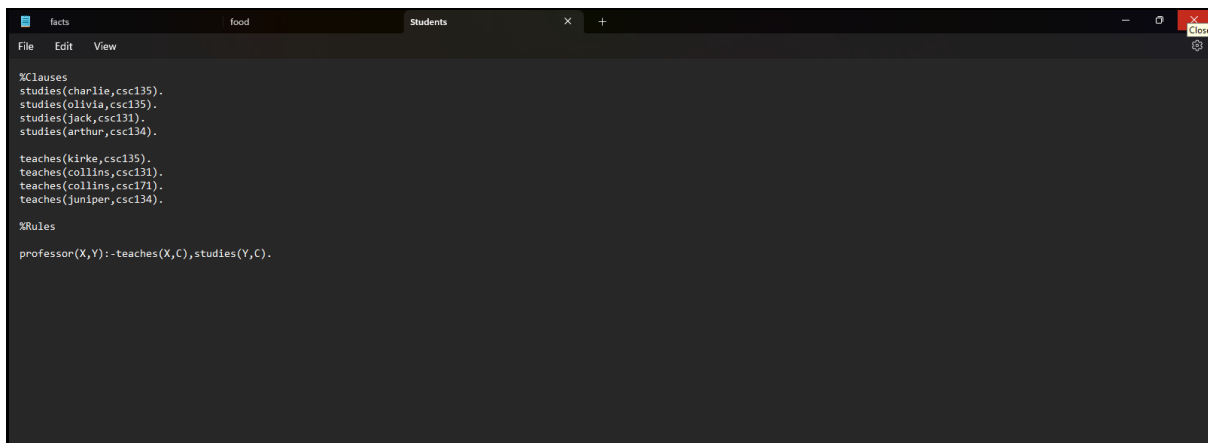
### AIM: Basic Prolog Problem

**Write a program in prolog to implement simple facts & Queries (Student, Teacher Relationship)**

### STEPS TO BE FOLLOWED:

Follow the same steps as above, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.



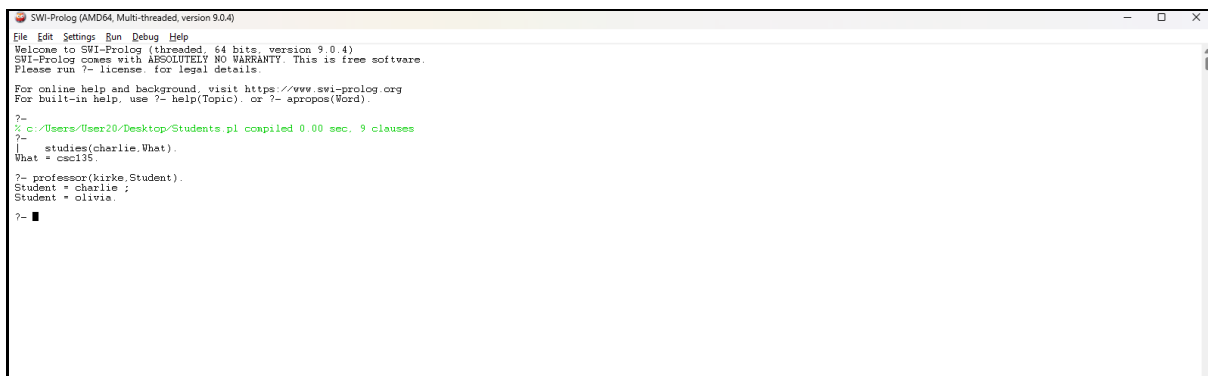
```
File Edit View

%Clauses
studies(charlie,csc135).
studies(olivia,csc135).
studies(jack,csc131).
studies(arthur,csc134).

teaches(kirke,csc135).
teaches(collins,csc131).
teaches(collins,csc171).
teaches(juniper,csc134).

%Rules

professor(X,Y):-teaches(X,C),studies(Y,C).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
?- c:/Users/User20/Desktop/Students.pl compiled 0.00 sec, 9 clauses
?-
?- studies(charlie,What).
What = csc135.

?- professor(kirke,Student).
Student = charlie ;
Student = olivia.

?-
```

## PRACTICAL – IV(d)

Date: 13/10/2023

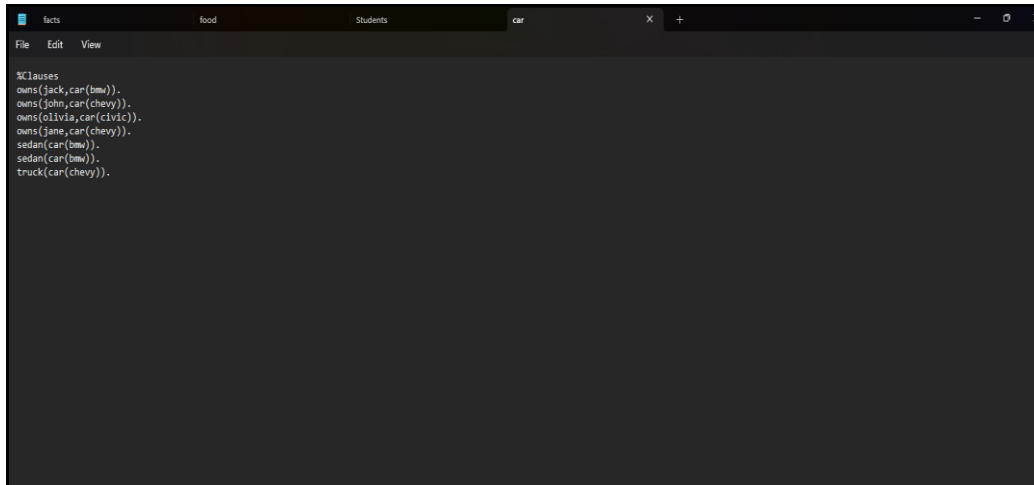
### AIM: Basic Prolog Problem

Write a program in prolog to implement simple facts & Queries (Car Problem)

### STEPS TO BE FOLLOWED:

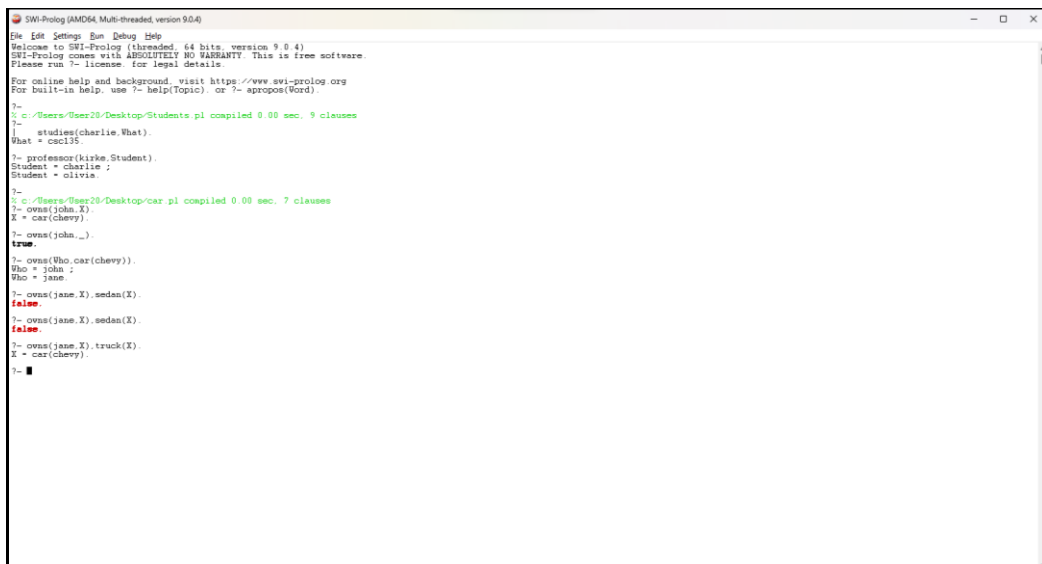
Follow the same steps as above, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.



```
facts
File Edit View

%Clauses
owns(jack,car(bmw)).
owns(john,car(chevy)).
owns(olivia,car(civic)).
owns(jane,car(chevy)).
sedan(car(bmw)).
sedan(car(bmw)).
truck(car(chevy)).
```



```
SWI-Prolog (AMD64 Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.
For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
?- c:/Users/User20/Desktop/Students.pl compiled 0.00 sec, 9 clauses
?- studies(charlie,What).
What = csc135.
?- professor(kirke,Student).
Student = charlie ;
Student = olivia.
?-
?- c:/Users/User20/Desktop/car.pl compiled 0.00 sec, 7 clauses
?- owns(john,X).
X = car(chevy).
?- owns(john,_).
true.
?- owns(Who,car(chevy)).
Who = john ;
Who = jane.
?- owns(jane,X),sedan(X).
false.
?- owns(jane,X),sedan(X).
false.
?- owns(jane,X),truck(X).
X = car(chevy).
?-
```

## PRACTICAL – IV(e)

Date: 13/10/2023

### AIM: Basic Prolog Problem

Write a program in prolog to implement simple facts & Queries (Pet Relationship)

### STEPS TO BE FOLLOWED:

Follow the same steps as above, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.

```
File Edit View
%Clauses
cat(fubby).
black_spots(fubby).
dog(figaro).
white_spots(figaro).

%Rules
owns(mary,Pet):-cat(Pet),black_spots(Pet).
loves(Who,What):-owns(Who,What).
```

```
?-
Warning: c:/users/user20/desktop/catrelation.pl:8:
Warning: Redefined static procedure owns/2
Warning: Previously defined at c:/users/user20/desktop/car.pl:2
% c:/users/user20/desktop/catrelation.pl compiled 0.00 sec. 6 clauses
?- listing(cat).
cat(fubby).

true.
?- listing(owns).
owns(mary, Pet) :-
    cat(Pet),
    black_spots(Pet).

true.
?- loves(Who,What).
Who = mary,
What = fubby.
?- owns(mary,_).
true.
?-
```

SIGNATURE: \_\_\_\_\_

# PRACTICAL – V(a)

Date: 14/10/2023

## AIM: Prolog Problem

Write a program in prolog to find maximum and minimum of 2 numbers.

### STEPS TO BE FOLLOWED:

Follow the same steps as above, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.

```
Compare
File Edit View

%Clauses
max(X,Y):-
(
X=Y -> write('both are equal')
)
;
X<Y ->
(
Z is X,
write(Z)
)
;
(
Z is Y,
write(Z)
)
).
min(X,Y):-
(
X=Y -> write('both are equal')
)
;
X<Y ->
(
Z is X,
write(Z)
)
;
(
Z is Y,
write(Z)
)
).
.-
```

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/Users/User20/Desktop/Compare.pl compiled 0.00 sec, 1 clauses
?- max(5,7).
ERROR: Unknown procedure: (-)/2
ERROR: is:
ERROR: [11] 5-7
ERROR: [10] max(5,7) at c:/Users/user20/desktop/compare.pl:4
ERROR: [9] toplevel_call(user: user: ...) at c:/program files/swipl/boot/toplevel.pl:1173
Exception: (11) 5-7 ?
% c:/Users/User20/Desktop/Compare.pl compiled 0.00 sec, 0 clauses
Exception details
Exception term: error(existence_error(procedure, (-)/2), context(prolog_stack([frame(11.call(user: user: ...), 5-7), frame(10.clause((clause)(000002033a20dd10), 8), max(5,7)), frame(9.clause((clause)(0000020339e20950), 3), toplevel_call(user: user: ...)))]), _70))
Message: Unknown procedure: (-)/2
In:
[11] 5-7
[10] max(5,7) erased clause from user: max/2 (no source)
[9] toplevel_call(user: user: ...) at c:/program files/swipl/boot/toplevel.pl:1173
Exception: (11) 5-7 ? abort
% Execution Aborted
?- max(5,7).
true.
?- max(-7,0).
0
true.
?- max(0,0).
both are equal
true.
?-
% c:/Users/User20/Desktop/Compare.pl compiled 0.00 sec, 1 clauses
?- min(2,3).
2
true.
?- min(0,0).
both are equal
true.
?-
```

## PRACTICAL – V(b)

Date: 14/10/2023

### AIM: Prolog Problem

Write a program in prolog to perform arithmetic operation

### STEP TO REMEMBER:

This time rather than saving the facts and clauses in notepad with “.pl” extension, directly write the operation which need to be performed in the CONSOLE.

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- X is 3+2.
X = 5.

?- X=3+2.
X = 3+2.

?- 3
|
ERROR: Type error: 'callable' expected, found '3' (an integer)
ERROR: In:
ERROR: [9] toplevel_call(user:user:3)
?- 3+2 is X
ERROR: Arguments are not sufficiently instantiated
ERROR: In:
ERROR: [10] 3+2 is _118
ERROR: [9] toplevel_call(user:user: ...) at c:/program files/swipl/boot/toplevel.pl:1173
?- X is +(3,2).
X = 5.

?- 5 is 3+2.
true.

?- 3+2 is 5.
false.

?- X is 3*2.
X = 6.

?- X is 3-2.
X = 1.

?- X is 2/2.
X = 1.

?- X is -(3,2).
X = 1.

?- X is 5-3-1.
X = 1.

?- X is -(5,3),1).
X = 1.

?- X is -(5,3,1).
ERROR: Arithmetic: '(-)/3' is not a function
ERROR: In:
ERROR: [10] _25076 is -(5,3,1)
ERROR: [9] toplevel_call(user:user: ...) at c:/program files/swipl/boot/toplevel.pl:1173
?- X is 3 mod 5.
X = 3.

?- X is 5^3.
X = 125.

?- Y is 3+2+4-1.
Y = 8.

?-
```

SIGNATURE: \_\_\_\_\_



## PRACTICAL – VI(a)

Date: 17/10/2023

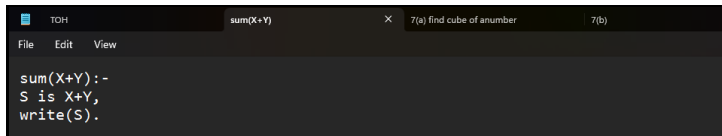
### AIM: Prolog Problem

Write a program in prolog to calculate sum of 2 numbers using variables.

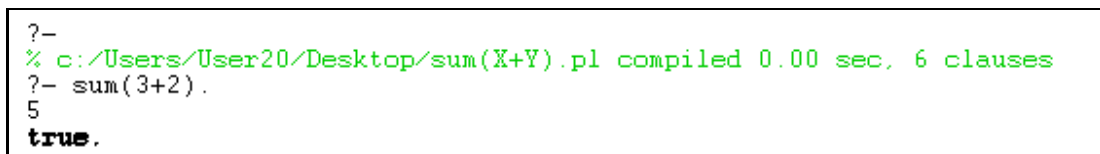
#### STEPS TO BE FOLLOWED:

Follow the same steps from the prior practicals, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.



```
sum(X+Y):-  
S is X+Y,  
write(S).
```



```
?-  
% c:/Users/User20/Desktop/sum(X+Y).pl compiled 0.00 sec, 6 clauses  
?- sum(3+2).  
5  
true.
```

## PRACTICAL – VI(b)

Date: 17/10/2023

### AIM: Prolog Problem

Write a program in prolog to calculate sum of 2 numbers and store in the 3<sup>rd</sup> variable.


#### STEPS TO BE FOLLOWED:

Follow the same steps from the prior practicals, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.



```
/*Another Way to Solve 6(b)*/  
sum(X,Y,Z) :- Z is X+Y.
```



```
?- sum(3,4,Z).  
Z = 7.
```

## PRACTICAL – VI(c)

Date: 17/10/2023

### AIM: Prolog Problem

Write a program in prolog to implement max(X,Y,M) so, that maximum of 2 number X and Y.

### STEPS TO BE FOLLOWED:

Follow the same steps from the prior practicals, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.

```
/*6(c)*/  
max(X,Y,M) :- X>Y,M is X.  
min(X,Y,M) :- Y>X,M is Y.
```

```
?- max(3,2,M).  
M = 3.
```

## PRACTICAL – VI(d)

Date: 17/10/2023

### AIM: Prolog Problem

Write a program in prolog to implement multi(N1,N2,R). where N1 and N2 denotes the numbers to be multiplied and R represent the result

### STEPS TO BE FOLLOWED:

Follow the same steps from the prior practicals, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.

```
/*6(d)*/  
multi(N1,N2,R) :- R is N1*N2.
```

```
?- multi(6,2,R).  
R = 12.
```

SIGNATURE: \_\_\_\_\_

## PRACTICAL – VII(a)

Date: 17/10/2023

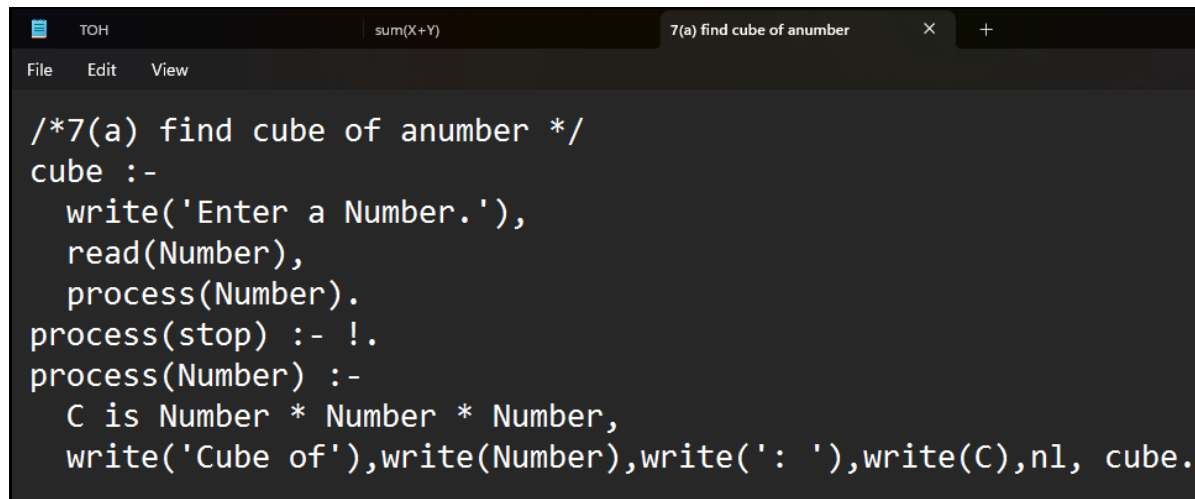
### AIM: Prolog Problem

Write a program in prolog to find cube of a number

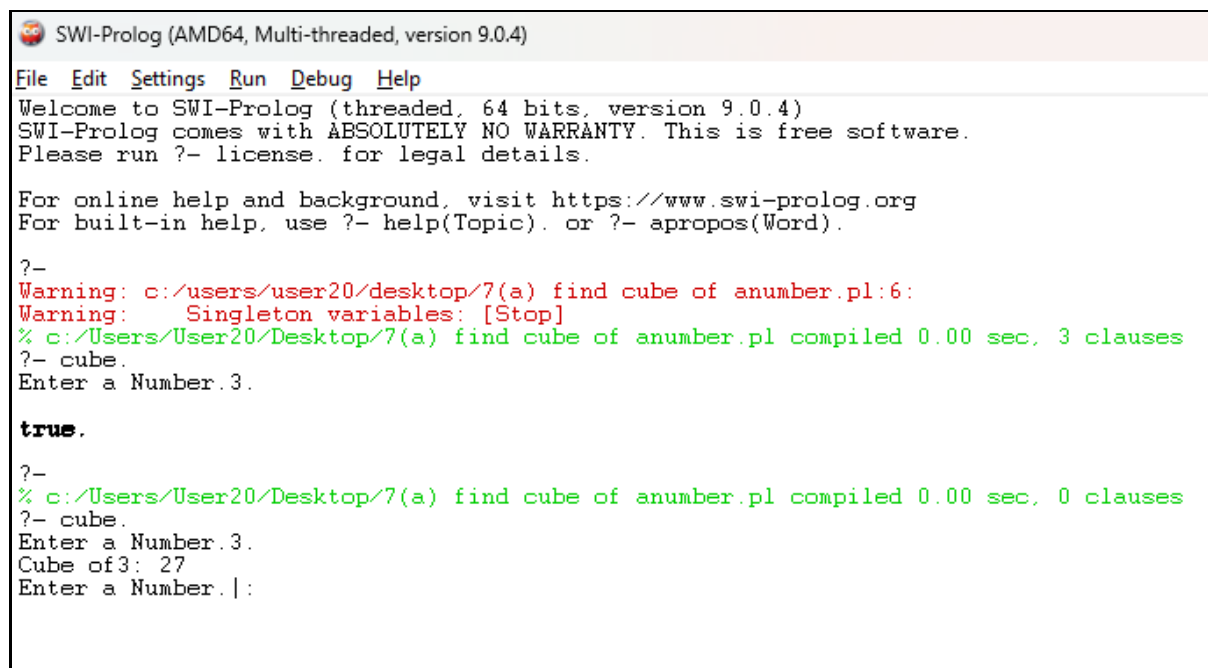
### STEPS TO BE FOLLOWED:

Follow the same steps from the prior practicals, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.



```
/*7(a) find cube of anumber */
cube :-
    write('Enter 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.
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
Warning: c:/users/user20/desktop/7(a) find cube of anumber.pl:6:
Warning: Singleton variables: [Stop]
% c:/Users/User20/Desktop/7(a) find cube of anumber.pl compiled 0.00 sec, 3 clauses
?- cube.
Enter a Number.3.

true.

?-
% c:/Users/User20/Desktop/7(a) find cube of anumber.pl compiled 0.00 sec, 0 clauses
?- cube.
Enter a Number.3.
Cube of3: 27
Enter a Number. |:
```

## PRACTICAL – VII(b)

Date: 17/10/2023

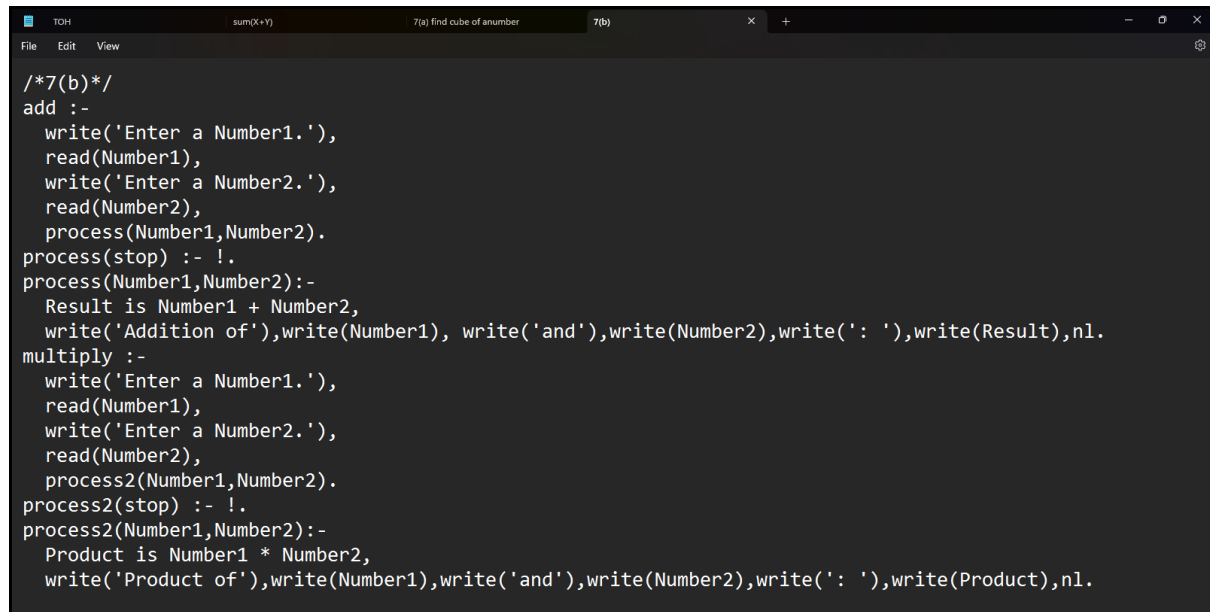
### AIM: Prolog Problem

Write a program in prolog to implement addition and multiplication by taking 2 numbers from user.

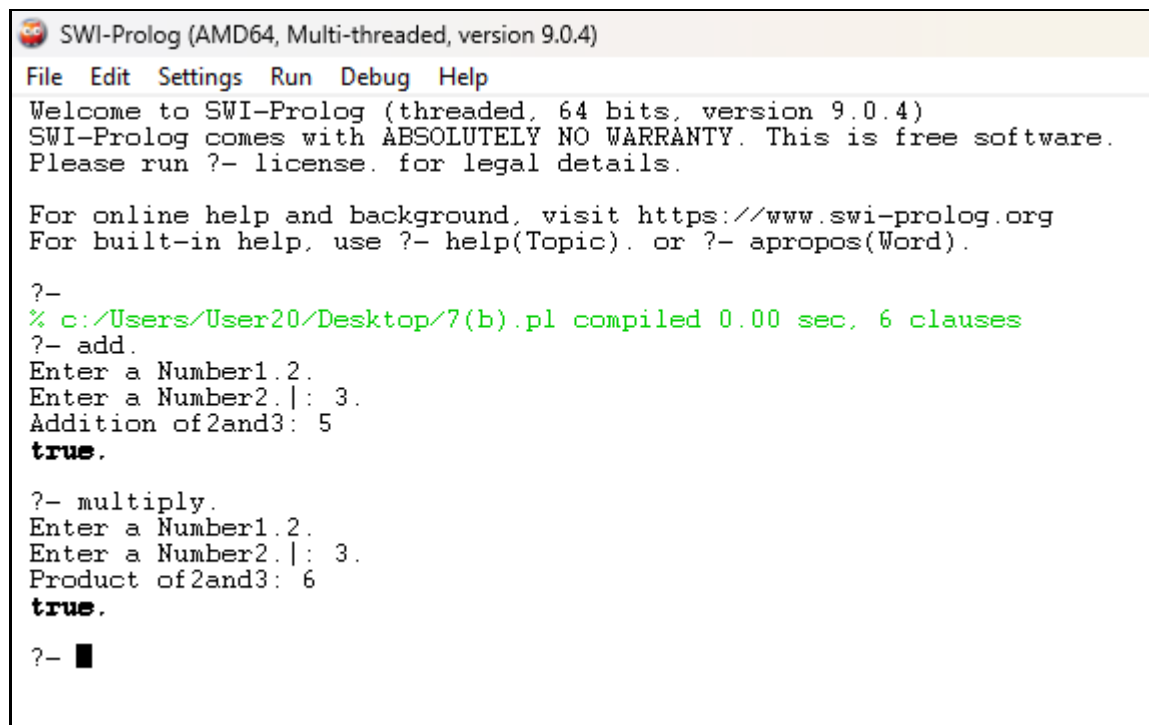
### STEPS TO BE FOLLOWED:

Follow the same steps from the prior practicals, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.



```
/*7(b)*/
add :-
    write('Enter a Number1. '),
    read(Number1),
    write('Enter a Number2. '),
    read(Number2),
    process(Number1, Number2).
process(stop) :- !.
process(Number1, Number2) :-
    Result is Number1 + Number2,
    write('Addition of '), write(Number1), write(' and '), write(Number2), write(' : '), write(Result), nl.
multiply :-
    write('Enter a Number1. '),
    read(Number1),
    write('Enter a Number2. '),
    read(Number2),
    process2(Number1, Number2).
process2(stop) :- !.
process2(Number1, Number2) :-
    Product is Number1 * Number2,
    write('Product of '), write(Number1), write(' and '), write(Number2), write(' : '), write(Product), nl.
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/Users/User20/Desktop/7(b).pl compiled 0.00 sec, 6 clauses
?- add.
Enter a Number1. 2.
Enter a Number2. | : 3.
Addition of 2 and 3: 5
true.

?- multiply.
Enter a Number1. 2.
Enter a Number2. | : 3.
Product of 2 and 3: 6
true.

?-
```

SIGNATURE: \_\_\_\_\_

## PRACTICAL – VIII

Date: 17/10/2023

### AIM: Prolog Problem

Write a program in prolog to implement Knowledge Base to Family Relationship

### STEPS TO BE FOLLOWED:

Follow the same steps from the prior practicals, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.

```
female(pam).
female(liz).
female(pat).
female(ann).
male(jim).
male(bob).
male(tom).
male(peter).
parent(pam,bob).
parent(tom,bob).
parent(tom,liz).
parent(bob,ann).
parent(bob,pat).
parent(pat,jim).
parent(bob,peter).
parent(peter,jim).
mother(X,Y) :- parent(X,Y),female(X).
father(X,Y) :- parent(X,Y),male(X).
haschild(X) :- parent(X,_).
sister(X,Y) :- parent(Z,X),parent(Z,Y),female(X),X\==Y.
brother(X,Y) :- parent(Z,X),parent(Z,Y),male(X),X\==Y.
```

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/Users/User20/Desktop/sum(X+Y).pl compiled 0.00 sec, 6 clauses
?- sum(3+2).
5
true.

?- sum(3,4,Z).
Z = 7.

?- max(3,2,M).
M = 3.

?- multi(6,2,R).
R = 12.

?- pow(2,3,P).
P = 8.

?-
```

SIGNATURE: \_\_\_\_\_

## PRACTICAL – IX

Date: 17/10/2023

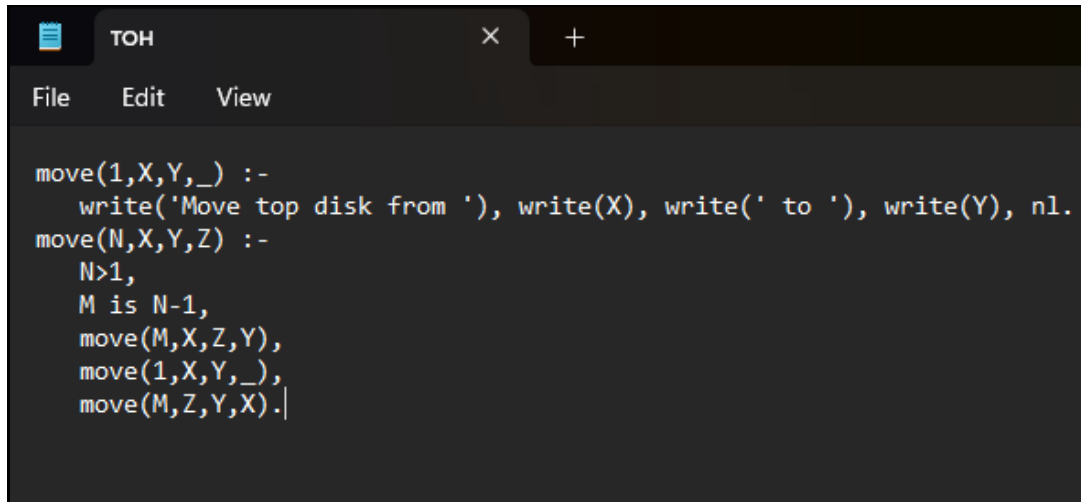
### AIM: Prolog Problem

Write a program in prolog to implement Tower of Hanoi

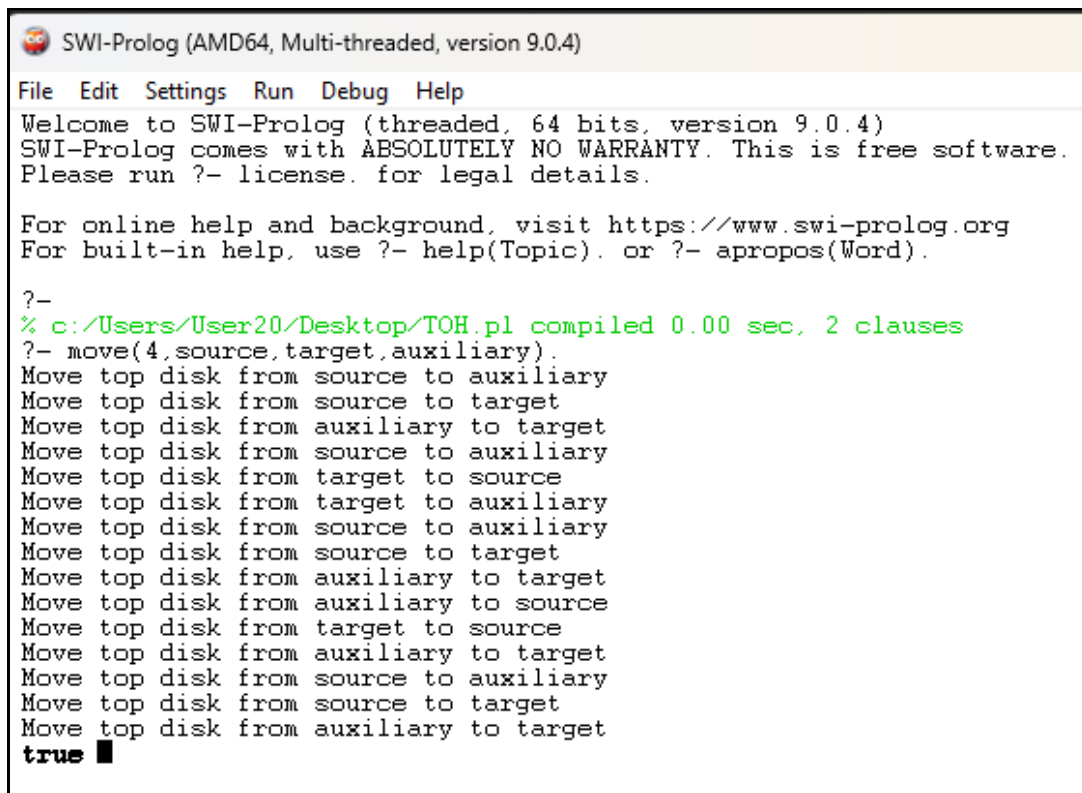
### STEPS TO BE FOLLOWED:

Follow the same steps from the prior practicals, except for the STEP 2.

This time save the facts, clauses and rules in the new notepad, save it with the “.pl” extension and upload at the time of querying.



```
move(1,X,Y,_):-
    write('Move top disk from '), write(X), write(' to '), write(Y), nl.
move(N,X,Y,Z):-
    N>1,
    M is N-1,
    move(M,X,Z,Y),
    move(1,X,Y,_),
    move(M,Z,Y,X).
```



```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
% c:/Users/User20/Desktop/TOH.pl compiled 0.00 sec, 2 clauses
?- move(4,source,target,auxiliary).
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
Move top disk from source to auxiliary
Move top disk from target to source
Move top disk from target to auxiliary
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
Move top disk from auxiliary to source
Move top disk from target to source
Move top disk from auxiliary to target
Move top disk from source to auxiliary
Move top disk from source to target
Move top disk from auxiliary to target
true
```

SIGNATURE: \_\_\_\_\_