#### **Program 1: Implement Tic Tac Toe**

```
17-11-23
      Implement Tir-Tac-Tol Grains
 board = [' . ] for a in songe (40)]
     insert other ( dotter, pos):
         board [pos] = letter
     Spore In Pace (pos):
         action board [pos] == 1
del paint Board ( board )!
   print 1 1 11 11 11 ) - 2 1 1 4 1 2 1
   + 1 + (= 1 boad + 1 1 + [17 boad + 1 1 + tries
     sport on boad (33). I tilly.
   paint ( 1 1 1 )
   point 1 11 + Loverd [4] + 11 + Loved [5] + 11+
           board [6]).
  paint ('111)
   printil " + Load (17 + "1" + hound (8) + "1" +
          board [9]).
dof in Linner (b; (1) in 1 and 1 and 1
  ( l = = [ [ ] d bna l = = [ [ ] and b[ ] = = [ ] newtor
  on (664) == I and brigg & and blod == 1) 09
(b[7] == 1 and b[8] == 1 and b[9] == 1)
( b == - [ ] d brag ( == C) d bra b == [ ] d)
(b[2)== l and b[5] == lbind b[8]== l)
(b[3]== l and b[4]== l and b[9]== l)
[1==[P]d bons l== [2]d bons l==[]d
(b[3] == 1 and b[5] == 1 and b[7] -= 1)
```

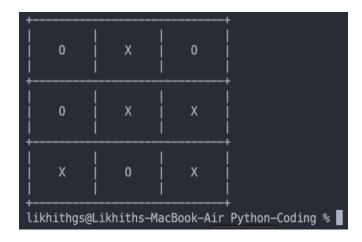
	Pags
	1
	det player More ():
	gun . True.
	while sun!
	more = input ( Enter position for X (1-9):1)
	day!
	mane = int (more)
	if more 70 and more (10'.
	H. Spore In Free (more);
	Sun : False
	jusert Latter ('X', more).
	else!
-	paint ( 'Spore in dicupied ).
7,11,	else!
64	. print ('Enter vrone in range').
	except:
7 19	paint ( ! Enter a number !).
	( I al believed
15)	def comp Move ():
4 1	nove: Franciscon, Francisconge (1,10)
160	Sun : dane,
	while sun:
	C)
	if (Space Inface (more)):
1 = + 1	insert Letter ('O', more)
80.	sun= false.
1:00	else! bud her fold has been to
do	Le Continue ! Continue !
. Ko	(heaf87d booth or f87d for he = (e)d)
10	Charpen hom see [ + 7d bus leo [2] d)
100	11 - 1-74 for 1 = 1271 for 1 = + 171d)
	1 PT 1 has bee 177 for bee 1574 )

Page
299 page major open 8
all mane):
while ( board, count ( ' 1) y's and board, count )
polyna i major service in model 3 describer 210)
player Mone Do
(brood (brood)
f ( & Winner ( board, 'x')):
point ( 'Human won')
bleak.
tata elso: 1 to 12 little
comp Move ().
paintBoad (board).
if (hwimer (board, 'O')):
point (' computer won')?
brook.
water and the state of the same was adopted to the
if ( board . ( ount ( ' 1 ) = = 0 :
point ('This is a draw').
100 labor 299 minutes de la laceration la constitución de la constituc
the same transfer that the same that the
all as a data de asset securit
with support of state of agreet, when the
grant and the same of the same
- Wall David Andrew
and mark being and
search see la

In the Beginning:

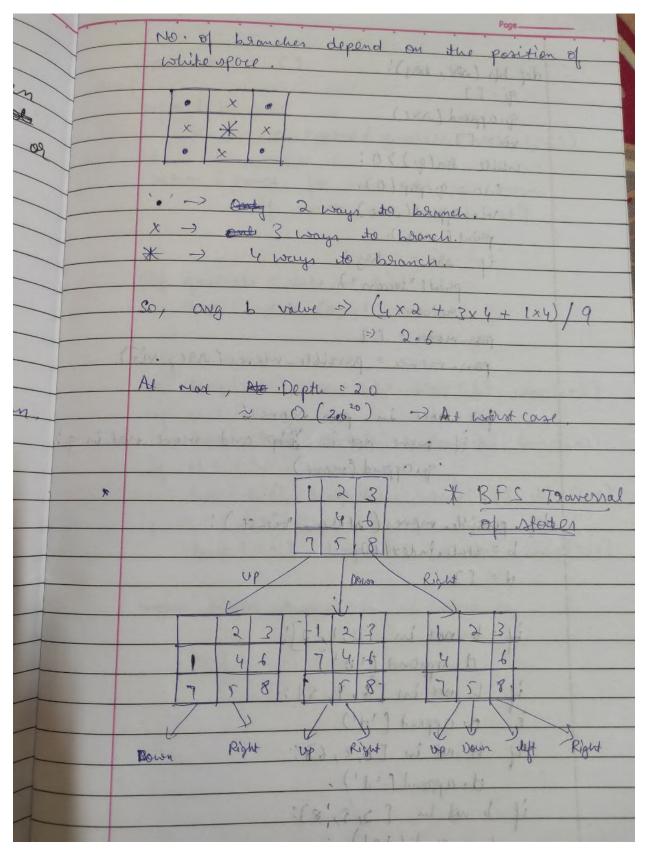
	nith G			096 7, 8,	9]	
	1		2		3	
	4		5		6	
	7	   	8		9	
com	outer':	s tur	n :			
	1		2		3	
	4		Х		6	
	7	     	8		9	
	r turn er a nu		on	the bo	ard :	

At the End:



# Program 2: 8 Puzzle Breadth First Search Algorithm

24/11/28	Page	
7 11 11/2s	8 Puzzlo Broklom volug BFS	1
*	Given \$ 3 x 3 Puzzle, where 8 places are number & I place in white space, biven 2 set of these puzzle, we have to sol	
	not.	
	Carlotte response to bring	
	Ex!	
	Juitial State Good state	
	1 2 2	
1	1 2 3 1 2 3 4 5 6 7 8 7 8	
	7 5 8	
	and and a second	
- 100	The white spore can more left, Dight, up, down,	-
Po	roposed rolution: using BFS, add all	
-	the sto new stokes formed, we will	
4	Ather stage of state & compare with	
G	order state. If they are some, then.	
1000	rint successful.	
(0	mplexity: O(b)	
F	> No. of brancher	
	-> depth.	
	P. T. 0	



by his ingression and he would be such a such as the s	
dot by (sex, targ):	
9:[]	300
grappend (sec)	1 1
My = [].	
while longy) 90:	Andto
(11c = 9. pop(0).	
vis append (ssac)	
point ( see ).	
: past == sea ;	
parlat ("success").	
netvin ! Detvin !	
populmones = []	
possimones: possible-mores (sec, vis)	
and the same of th	
for more in posimoves:	
if more not in wip and more not in q!	
'grappend (more)	
3000 mp 2 2 2 3 4 8 8 8 9 9	(1-15)
olef possible-mores (staly, vis-ct):	
b: state, index (6)	
d = []	
if b not in [0,1,2]!	
d. anoma (1111)	
d. append ('U').	
A MA M FOLLOW.	
t d'append ('d')	
if b not in [0,3,6]!	1
d. append ('d')	1
if b nd in [2,5,8]:	
d. expend (191)	

	The state of the s
	pom-mones = []
rudis	Agail pringsoft subsoct to loss of
	foreign and lett of it mesages were
	pos mones, append (generate (state, 1, 5))
no str	Sodie [ state / 1/ b))
	repurs [ moves for moves in pass-moves.
	H manes and it
	if moves not in virast?
	Company and I am about a all the
	aled accepted a date on the little bit got
	Obj generate ( state, m, b) i esté segui
	temp : state: copy () stage of lab
	1.0 = sityot. fi
	y m== d'! weight
	temp[b+3], temp[b] = temp[b], temp[b+3]
	if m == 1V1: garage where
9:	temp[6]; temp[6]: stemp[6], demp[6-])
	if m = soldie mt ton man fi
(i-stret )	demp[b-1], temp[b] = demp[b] temp[b-1)
	ig me = 'd'istage there ji
	temp. Ebtity, temp[bt? temp(b), temp[bti]
	Homb Bill I wanter
	the state of the state of
	rece detant temporation at other las
	( Hyph, Salgery ) Ab = other depths).
	toware to
	Aug B viewlorg

```
Likhith GS-1BM21CS096
1 | 2 | 3
4 | 5 | 6
0 | 7 | 8
1 | 2 | 3
0 | 5 | 6
4 | 7 | 8
1 | 2 | 3
4 | 5 | 6
7 | 0 | 8
0 | 2 | 3
1 | 5 | 6
4 | 7 | 8
1 | 2 | 3
5 | 0 | 6
4 | 7 | 8
1 | 2 | 3
4 | 0 | 6
7 | 5 | 8
1 | 2 | 3
4 | 5 | 6
7 | 8 | 0
```

success

Program 3: 8 Puzzle Iterative Deepening Search Algorithm Observation:

A  12  23   Date   Page   Pa
8 Ruzzla Parollem using ID DFS
The second secon
The goal of Itorative Das pening Dopth first Secon algorithm in to iterate Dopth wise,
Sever algorithm is to iterate super wise
is level by level.  At lock level, we apply DFS for that
as a limit.
· Parada da dan onen 13
def id-dfr (puzzle, god, ga-mover):
short iter tools
def der (Franke, depth).
if 'dopth = = 0'.
Nester.
if 200te[-1]== gool: [24]
Japan Sante
for more in get mover (soute [-1]):
if more not in route!
near soute: de l'aoute + [mone ], depth-1)
if next-sporte!
strong tryn muter
for depth in iterbook. count()!
soule = dp ( [puzzle], depth).
if roubl!
return route
meron order.

del possible mont
def possible moves (state):  b = state index (a).
d = []
(a de la como la la como de la co
d. append ('v')
if south : [8,7,8]: 100 d fi
I and append (1'd1). I am 12 1 14 14
if b not in 10,3,67:
d. append (11)
if 6 not sin [2, 5, 8]!
d'append ('a')
E] : money > []
for i in d:
pas-mones append (generate (state, 1,6))
return por-mover.
det generate (votate, m, b):
temp = State, copy()
H m == 'd':
demp[b+3], demp[b] = temp[b], temp[b+3]
if w = 2 'U'!
tomp [b-3], temp[b] = temp [b], temp [b-3]
F1 M = = 1111
temp[b-1] temp[b] = stemp[b], temp[b-1]
1011
temp[b] = temp[b], temp[l
The state of the s
return tomp.

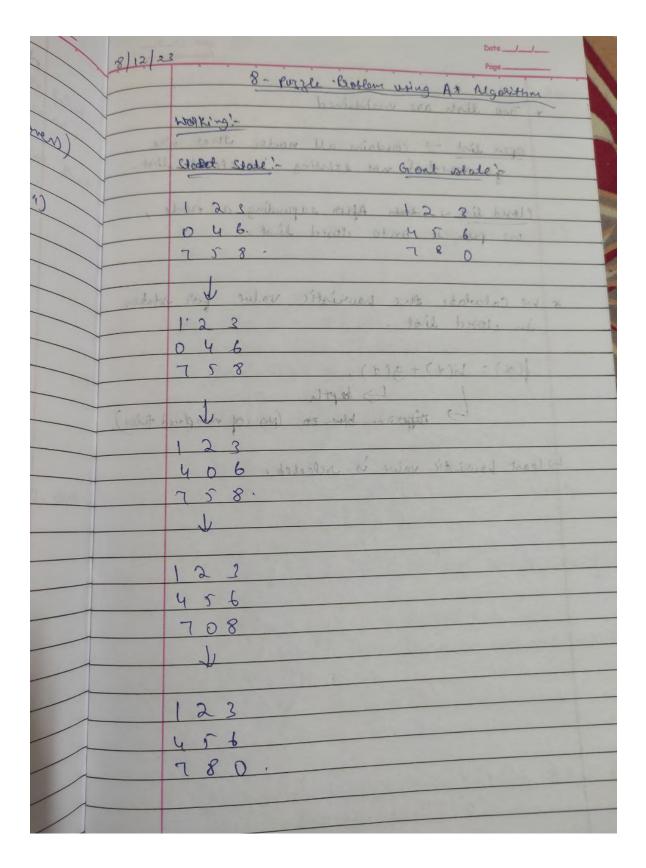
-	
-	initial = [1,2,3,0,4,6,7,5,8].
	goal = [1,2,3,4,5,6,7,8,0]
	sioure : id of (initial, goal, possible movers)
	to the the sage of
	if soute! 18 1 17 11 ton of 1
	("surfar bet which of it is mouse") tuning
	point ("Path!", noute)
	else!
	paint (" Foiled to find Solution").
	(ce) beggand
1	
	Ed swar sag
	ih wi i sal
601	I tota I elacuso I hayyo conservacy
	BUMPY OF MARKE

Likhith GS-1BM21CS096

Success!! It is possible to solve 8 Puzzle problem

Path: [[1, 2, 3, 0, 4, 6, 7, 5, 8], [1, 2, 3, 4, 0, 6, 7, 5, 8], [1, 2, 3, 4, 5, 6, 7, 0, 8], [1, 2, 3, 4, 5, 6, 7, 8, 0]]

## Program 4:8 Puzzle A\* Search Algorithm



Page
marin 14 th aview modeld - 1000 - 8
to list are maintained
- Constitute
open list -> contains all nodes that we
generated & not existing in closed list
closed list > low After eapunding a node we push it on to closed list.
me push it on to closed list.
* We Calculate sue previstic value for states
in closed list.
4 4 0
f(x) = h(x) + g(x).
bepth misponer blue to (no of misplaced tiles).
5 6 1
4) Least housistic volve in selected.
.87
122
805
2.5
+ 7 V

# Program 5 : Vacuum Cleaner

Vaccioni C Da
Soulstate = 1 . 1' ! D' 12'
WINDOW CALL TO A CONTROL OF THE CONT
cost = 0.
point( " Enter the Glasting location of Bot (A(B)")  point()
docation - the Charting location of a
docation = input() point()
for som in sance
too 200 11 11 12 12 12 12 11
for som in sonstate:
Groom Chate [ Goom ] = oction.
if soomstate Is an ection.
if soonstate ! = goulstate :
H location == 'A':
if agam State FLAIR
if sometate [1A1] == 1/1: # A h disty
ट्रांची ने - विकास में निर्माण के
print ("Lor & won Disty and now cleaned")
if soom state : goal state :
point (" Groat State has been met "+ ctal Got)
else 1 total has been met + challoot)
thelse !- with I and A wook for
point ('In A -> B")
and there is contitoole with head is should
& mage
if tonomichale [18'] == 11': #R in diry
and the state of t
9.00m State [13'] = 0'
Lan mate costores how have is seen
point ("Loc 13 was Dirty and now cleaned ").
toned ").

if docation == '13'! of growing ater 18'] == "1" # B is die 300mdate ['B'] = '0' paint (" LOCB was disty non clean ") Snoom State = = goal state! print ("Goal state achieved " of stor (con) paint (" In R -> A"). (Off += ) if Droom State ['A'] :: 'L' # A L dien 10° = (1 A 1) = D1 1000 to 1000 point (" Loc A was dirty now clean") if soon State = 2 goal State! point (" Copal State orlivered " + chilcon) ul I else ! point (" All rooms are already clean") point (" Total cost " + (ost) \* Logic!check of mittal state == Good state of Not For soon A', ther if dirty -> cost +1 if clean -> No cost Check if Goal Make reached, If not be to Doom B. In soon 131 1- if disty - Clean it. if cloon -> No lost. check if Groal state realed, If not too to resid

Vaccour Cleaning Agents -> 4 scoons Spalstate = 2'11'10', 181:10', 101'6', 101:101 Deconder & A! !'O', 181: 10; 10': 10', 10': 0' cont = 0. point (" Enter stocking location of Bot XAIBIC 104") docation: input() point ) for room in room State: Tropon State Toron ) . Input (" Enter State") while Grown State 1 = Goal State! 4 location in secon state! cula\_ drown : do cation. H. Iroan Stal [ cult hoom) = = 111: cost 11. nest. 900m = input ("Enter") if next soon in soon stake and next sucon ! = white cost tel. Location: rest- rosen point ( " lost : ", cox ) Point (" Groat chate has been Achiewed ) n") paint (" leg formance Manuali"; cost)

```
Likhith GS-1BM21CS096
Enter clean status for Room 1 (1 for dirty, 0 for clean): 1
Enter clean status for Room 2 (1 for dirty, 0 for clean): 0
Cleaning Room 1 (Room was dirty)
Room 1 is now clean.
Room 2 is already clean.
Returning to Room 1 to check if it has become dirty again:
Room 1 is already clean.
Room 1 is clean after checking.
```

#### Vacuum cleaner 2 rooms

```
Likhith GS-1BM21CS096
Enter clean status for Room at (1, 1) (1 for dirty, 0 for clean): 1
Enter clean status for Room at (1, 2) (1 for dirty, 0 for clean): 0
Enter clean status for Room at (2, 1) (1 for dirty, 0 for clean): 1
Enter clean status for Room at (2, 2) (1 for dirty, 0 for clean): 1
Cleaning Room at (1, 1) (Room was dirty)
Room is now clean.
Room at (1, 2) is already clean.
Cleaning Room at (2, 1) (Room was dirty)
Room is now clean.
Cleaning Room at (2, 2) (Room was dirty)
Room is now clean.
Returning to Room at (1, 1) to check if it has become dirty again:
Room at (1, 1) is already clean.
```

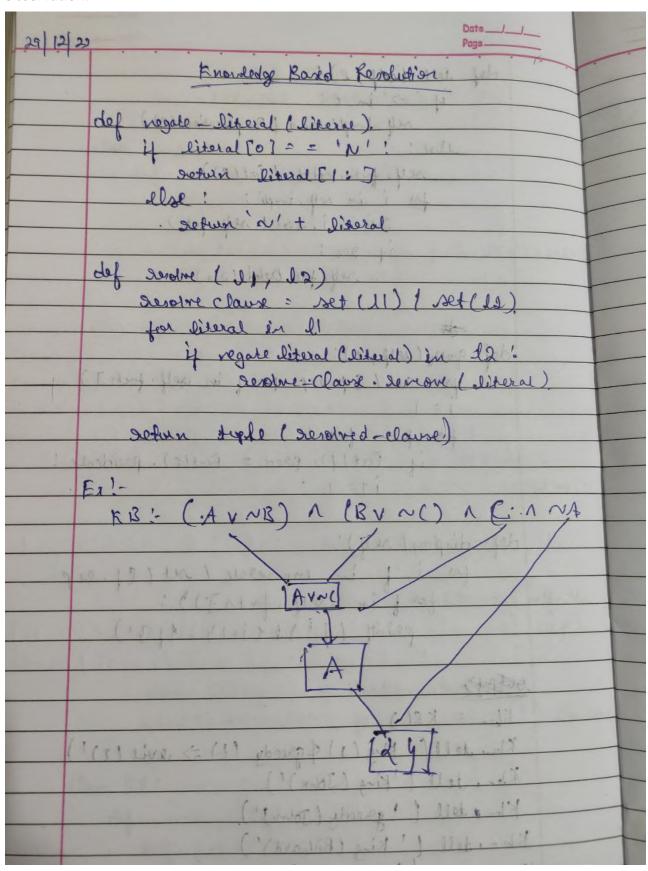
#### Vacuum cleaner 4 rooms

# Program 6 : Knowledge Base Entailment

ay let as
Kyonsledge P.
from sympy Import symbols, And, Not, Implier,
satisfiable import symbols and as
del create- KG ()
Carate Ki C.
satisfiable import symbols, And, Not, Implier,  def create-Kb():  p: symbols ('p')
2 c aymbols ('p')   free age
And implies to
( (a) (a) to harptient qua) storica )
Solvan Kb
det queig entails (Klog group):
entailine ( Riby green) is recorded
entailment = Satisfiable ( And ( Kb, Not (growy))
Between Not entailment. (And (Kb, Mot (query))
function: -how ( ) demand on the
4 MARCO PARTICIPATION A
alel )
Hairs function: - hours ( ) can less ( implies or not.)
og main(): : ( A) The I who job
og main(): : ( A) The I who job
Kb: Create- Kb(1.)
Kb = Create - Kb(1.)
Kb = Create - Kb(1.)
Kb = (reate - Kb(1).  (query = Symbol (1pl) makes  idesult = query-entails (Kb, query)
Kb: Create - Kb(1).  (query = Symbol (1/pl) make a  idesult: query-entails (Kb, query)  Reint (11 Knowledge Rose: 11, 1848)
Kb: Create - Kb(1).  (query = Symbol (1/pl) make a  idesult: query-entails (Kb, query)  Reint (11 Knowledge Rose: 11, 1848)
Kb: Create - Kb(1).  [query = Symbol (   pl)   plant    [sesult: query-entails ( Kb, query)  [sesult ( " Knowledge Rose!",   kb)    [sesult ( " query ( " query )   plant   " query )
Kb: Create - Kb(1).  [query = Symbol (   pl)   plant    [sesult: query-entails ( Kb, query)  [sesult ( " Knowledge Rose!",   kb)    [sesult ( " query ( " query )   plant   " query )
Kb: (reate-Kb(1)  query & Symbols ('pl) mayor  sesult: query-entails (Kb, query)  paint (" knowledge Rose:", kb)  paint (" query:", query).   100  paint (" query:", query).   100  paint (" query entails (knowledge Rose:", nearlt)
Kb: (reate-Kb(1)  query & Symbols ('pl) mayor  sesult: query-entails (Kb, query)  paint (" knowledge Rose:", kb)  paint (" query:", query).   100  paint (" query:", query).   100  paint (" query entails (knowledge Rose:", nearlt)
Kb = (reate_Kb().  (query = Symbolar(!pl) mare = e  idesult = query-entails (Kb, query)  print (" Knowledge Rose!",   kb)  print (" query!", query).   print (" query entails (Knowledge Rose!", nearlt)  [Dagic 1) march   march   march   march   meanlt)
Kb = Create - Kb(1.)  (query = Symbola (ip))  Desult = query entails 1 Kb, query)  paint (" Knowledge Rose:", 1kb))  paint (" query: " query) 1912  print (" query: " query) 1912  print (" query entails "Knowledge Rose:", nearlt)  Logice L mas (1913) des (1914) from (1914)
Kb: Create Kb(1)  (query & Symbolistical TKb, query)  print (" Knowledge Rose:", 186)  print (" query: ", query), 196  print (" query: ", query: ", query: ", pend)  print (" query: ", query: ", query: ", pend)  print (" query: ", query: ", query: ", pend)  print (" query: ", query: ", query: ", pend)  print (" query: ", query: ", query: ", pend)  print (" query: ", query: ", query: ", pend)  print (" query: ", query: ", query: ", pend)  print (" query: ", query: ", query: ", pend)  print (" query: ", query: ", query: ", pend)
Kb = (reate_Kb().  (query = Symbolar(!pl) mare = e  idesult = query-entails (Kb, query)  print (" Knowledge Rose!",   kb)  print (" query!", query).   print (" query entails (Knowledge Rose!", nearlt)  [Dagic 1) march   march   march   march   meanlt)

```
Likhith GS 1BM21CS096
Knowledge Base: ~r & (Implies(p, q)) & (Implies(q, r))
Query: p
Query entails Knowledge Base: False
```

### **Program 7: Knowledge Base Resolution**



```
Likhith GS 1BM21CS096
        |Clause |Derivation
Step
1.
        | Rv~P
                | Given.
          Rv~0
                  Given.
2.
3.
          ~RvP
                  Given.
          ~RvQ
                  Given.
4.
5.
          ~R
                  Negated conclusion.
                  Resolved Rv~P and ~RvP to Rv~R, which is in turn null.
6.
A contradiction is found when ~R is assumed as true. Hence, R is true.
Likhith GS 1BM21CS096
```

```
Step
        |Clause |Derivation
1.
          PvQ
                  Given.
          ~PvR
                  Given.
 2.
          ~QvR
                  Given.
 3.
4.
         ~R
                  Negated conclusion.
          QvR
                  Resolved from PvQ and ~PvR.
 6.
          PvR
                  Resolved from PvQ and ~QvR.
          ~P
                  Resolved from ~PvR and ~R.
                  Resolved from ~QvR and ~R.
 8.
          ~Q
                  Resolved from ~R and QvR.
          Q
 9.
 10.
                  Resolved from ~R and PvR.
                  Resolved from QvR and ~Q.
 11.
                  Resolved R and ~R to Rv~R, which is in turn null.
 12.
A contradiction is found when ~R is assumed as true. Hence, R is true.
```

```
Likhith GS 1BM21CS096
         |Clause |Derivation
Step
 1.
          | PvQ
                   | Given.
           PvR
                     Given.
           ~PvR
                     Given.
           RvS
                     Given.
 4.
 5.
           Rv~Q
                     Given.
                     Given.
           ~Sv~Q |
 6.
           ~R
                     Negated conclusion.
                     Resolved from PvQ and ~PvR. Resolved from PvQ and ~Sv~Q.
 8.
           QvR
 9.
           Pv~S
          | P
                     Resolved from PvR and ~R.
 10.
           ~P
                     Resolved from ~PvR and ~R.
 11.
 12.
          | Rv~S |
                     Resolved from ~PvR and Pv~S.
 13.
                     Resolved from ~PvR and P.
          | R
 14.
          i s
                     Resolved from RvS and ~R.
 15.
          | ~Q
                     Resolved from Rv~Q and ~R.
 16.
           Q
                     Resolved from ~R and QvR.
           ~S
                     Resolved from ~R and Rv~S.
 17.
18. | Resolved ~R and R to ~RvR, which is in turn null. A contradiction is found when ~R is assumed as true. Hence, R is true.
```

# Program 8 : Unification

MAIN Date Page
19/1/24 Unification
del get Attributer (enp):
one : ore. selit ("C")[1:]
exp = "(", john teap) : (137 - 1000) 100
eap = se. split ("(? +1) (.))" [eap)
setuan exp. ('e') radings
action appoint (""") [6]
dit wester
def is const (char):
noturn charingper() and len (char) == )
(Complete and the A & Wholfield = translation
def in val (char). Hour listers, take white
Seturn char, indower () and len (char) == 1
det apply less, sub):
to so In July of so In
l = deploce Alphillure ( e, o, neus)
Sepange at the land the sample
def unity (expl , "expl 2 ) balward " ) was
if ( eapl 5 paroap 21) week " I wise
June ( " query entains [ Grandtage Book! " ) mand
if is court (exp1) and be court (exp2)
- ( 7 C-2, 7 C- h) prefugne-Falogs. (- ) such es
(Bosically implies).

Lore
4 is (oust (eapl):
Defun [ conel a sol ?
TOTAL PRODU
solven [(eap2, eap1)]
Ct-1 \ \ \
A STORAGE SALES
their thun (exp), ones):
setun False with a formalist
telet: + [i: 7tot) + 'T' = tota war
return [ loap 2, eapl ) ] [ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
T + 1 1 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2
if in the get Trit fred (exp1) 12 act Trit good (exp).
part to Flatch of Dedicates "), 2 has
Defusion False? 3000 12 1000
ac1 = len(get Att (ox f1))
acz = lon (get Art (eapz)) : talk
if all 1: al2 ! - Herometer different
an ende: " Drokeni False war + Fed 172 = 2
initSub = unify (getfp (dap), getfp (dapa))
('Hu') rebuistoss = 1
4 acl==1: 1 total 1 to = 1015
Sweet nitsot. (3) was a late
The state of the s
init Sub + extend ( somewh) walnut total
greturn & trit get & Jos 1) wayse , tot2 = 60 to
1/1F  M71/ 2 Kare
esel = knows ( * X) listerifice = too
Ung 1 and 100
eap 2 = "known ( Richard)"
Sub : unity ( exp 2).
point ( SOK)
The following of the state of t
astent! [('x', 'Richard')].
output? [(X), Richard)].

```
[5] exp1 = "knows(X)"
    exp2 = "knows(Richard)"
    substitutions = unify(exp1, exp2)
    print('Likhith GS 1BM21CS096')
    print("Substitutions:")
    print(substitutions)
    Likhith GS 1BM21CS096
    Substitutions:
    [('X', 'Richard')]
    exp1 = "knows(A,x)"
    exp2 = "knows(y,mother(y))"
    substitutions = unify(exp1, exp2)
    print('Likhith GS 1BM21CS096')
    print("Substitutions:")
    print(substitutions)
    Likhith GS 1BM21CS096
    Substitutions:
    [('A', 'y'), ('mother(y)', 'x')]
```

# Program 9 : FOL to CNF

19/11	Acces 3-20 Date 1
-14-4	FOL to CNF (grad) tours
	1 (Care 1 200) I mine
	ingot seet together
	Tigo Guel mobile
	def fol- to- ent (fol):
	while 'I have statement in the
	i = statement. Indea (!_!)
	now - stat = ' [' + stat [: 1] + " = >" +
	State 1: ] + [] ( + ( + ) = ( + ) + (
	Stat[:i] + ']'
:(6	· if inter Oct Till Pard ( on p 1) 10 gol Till Pard ( on p
-	too s in whater of Partet in 2 long
- 1	s = stat s. sopore (s, fol to cy (s))
-	nets and get her (orgi)
	while '- ' In Atat : ((Cgre) teA top) lieb = 610
	if all a all ('-') thebrul its shiften
	three sale: O(Bd fi) popu-wen + [ed:]2 = 2
	while I with in stat ? 97/20) I fore = July 1
	i= Not index ('ww')
	21 at : list ( Mat ) : 1 = 2   20
	state 's join (2) wished with
	Stat = stat. suploce (In [+", In+")
	100 = Stat. Supplies (1 N [ ] / ) 1 / ) 100
	(:E1 A 1), s chea
2	tot = De. findall Keiph Mat ).
	Chamber Duant a case
	Can bile 120 Con Not
	The standing

V	tor 5 in material Page
1	C) . Matatage
	State : Se findale ( second (C))
1	State 2 01 - 101 -
1	se findale (
1	State : De findale (peps , star).
tiles	for S in Stabin
33045	The soll is
	selven state selves (5) Dellas
	selven state (s. Derrorgon (s))
	paint 1 cm
	(Skolenization 1 top 10
	paint (Skalenijation (fol to ent ("animaly)))
	point (Skolenijation ( 4.4)")))
	(u)
	point (fol. to con 1" [american (x) & weapon (x) & sells  (x, y, z) & horrise (z) [2) (2) (2) (2) (2)
- 1	Language Company
eles bus	( hortile ( 2) 7 Wagen (y) & sells
9	output?
100	output?
1 1.14	[ ~ animal ( ) )
	[ naminal (y)] dover (a, y)] & [ Nover (a, y) ]
·lla	town (a,y)
1	Parinal (Gara)
• [	~ american (2) [ ~ weapon(y)   ~ wells (2,4,2)   ~ horitor
	weapon(y)   rule (x, y 2)   and
10	and presimination (g) ( wells (x,y, 2) / whomen
- 1	and won it for thousand top it now
14	SERC'S
Ta	
50	by -
-	class F.P.
Roy	and Andrilla to the toning the
1	nove quantifiers by scholamijation
Spli	I she sanger is
it	. Is found & rolling
App	ly Perredgan.
Color	intil one 1

```
print(Skolemization(fol_to_cnf("animal(y)<=>loves(x,y)")))
print(Skolemization(fol_to_cnf("∀x[∀y[animal(y)=>loves(x,y)]]=>[∃z[loves(z,x)]]")))
print(fol_to_cnf("[american(x)&weapon(y)&sells(x,y,z)&hostile(z)]=>criminal(x)"))

Likhith GS 1BM21CS096
[~animal(y)|loves(x,y)]&[~loves(x,y)|animal(y)]
[animal(g(x))&~loves(x,G(x))]|[loves(F(x),x)]
[~american(x)|~weapon(y)|~sells(x,y,z)|~hostile(z)]|criminal(x)
```

# **Program 10: Forward Reasoning**

19/1/24  LAB-ID  FORWARD REASONING  Clan Implication:  del - init- (self, exp):
LAB-ID FORWARD REASONING  Clan Implication:
Clan Implication:
Clan Implication:
Clan Implication:
and a - order of wary of the
selfe eap = exp
J: 010 ARIH ( 12)
self, line = [ Foot (f) for f in 100 mes
self. Als = Fort (151)
def evaluate (welf, tacks):
court = RN 1/ port man.
water the
Clar should to it forth and to the
for val in self. des !!!
if ( pled case : f. predicate :
for i, v. In enumerable (val-get toly o)
y! - ! toptor
Culphandra 3 [ Mr. ) and Boundard (V) 1 = f. getland () [i].
TOP tombre
pleal, attr' = get fall dicate ( self the
(gas famul (com o lower o tours)) [ Jan (P)
and I commenced to I compared to 1 commenced.
actorn foot (expr.) if then (manifus) and
all (f. get Result () for fin new-lbs]) else More
militinga resoul for the new sans sea mil
1000
clan KB!
dof ivit (self):
self. fortr : set ()
self. impli = set()
and much a source
medowish Back !

	the self (self e):  Page
	Page
	self implication
	Use: Josephinghing (Implication (P))
1	self forth, and
	for im sell (fact (P))
espir	Dies of bright;
Aug)	self forth end ( Fort (P))  Jos : Level ( self fort)
1	solf, foren Polici (ses)
	( Jes )
1	dala
1	def growy (self, e):
1	the seal fabreau
-	too it is eat tout in well tour ]
-	for it in foch!
1	in Fort 11)
00).	if fort 11). gred = fort(e). preideas:
	def display ( self)!
	for i fin enumerale ( set ( et eleg
	print (11) + (1+14. 4131).
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	200 feet or
4	Kb== KB()
	Kb. tell (' King (3) & greedy (1) > revis (2)!)
	Kbc. tell ('King (John)')
	Kb- e tell ( 'greedy (John)')
\x	ib tell (' King (Richard)')
V	My (Mynara)
h	b - griery ('evil (1)')
- protect?	
0	eling evil(1)!

```
Likhith GS 1BM21CS096
     Querying criminal(x):

    criminal(West)

     All facts:

    criminal(West)

    enemy(Nono,America)
    owns(Nono,M1)
    missile(M1)

               weapon(M1)
               hostile(Nono)
               7. sells(West,M1,Nono)
               8. american(West)
[4] kb_ = KB()
     kb_.tell('king(x)&greedy(x)=>evil(x)')
     kb_.tell('king(John)')
     kb_.tell('greedy(John)')
kb_.tell('king(Richard)')
                                                                          Focus the last run cell
     kb_.query('evil(x)')
                                                                          02:05 (0 minutes ago)
                                                                          executed in 0.008 s
     Querying evil(x):

    evil(Richard)

               evil(John)
```