	Page
QI	Write a program for Tic-Jac-100 games
	import Aandom
	tie=[1,2,8,4,5,6,7,8,9]
	def point Board (trie) has branch a shorm () XXX
	print (Hic Co]+ (1'+tic[i]+(1'+tic[2])
	Position (" 1-1) - 1 - 1 - 1 - 1 - 1 - 1 - 1 -
La 60.	point (tic[3]+11/+ tic[4]+4/+tic[5])
	porcial ("
	porced (tic(6)+41+tic(7)+11+tic(8))
	since assignate specialist (1) (0)
	det iswinier (tic, pos) marenge (1)
	I Salance Gacyray
	of tic [0] == Hic [4] and tic [4] == tic[8] or
Roat 2	fic[2]==fic[4] ad fic[4]=fic[6]
	Stephen Tours Tour Strate Air Food - 37 =
	else if tic [poso] = tic [pos-3] and tic [pos-3]== tic [pos-6]
	else if tic[pos //3+1]=tic[pos //3+2] and tic[pos/ 3+2]
, plu	3-3 3-3
	Gebre True
	gester False
	the applate uses (Hic):
	pu = int (input (" Entra a number on the board"))
	while (num not intic)!
	pur = int Compute (" Cite a mill or the board
	tic [nu-1] == '0'
	dy update - ca p(+ic):
	for i in tie:
	of i! = 'x' ad i! = 'a':
<i>M</i>	4(15-1)=(x,1-1)= 7me).

	Page
	OBSTITUTE OF THE STATE OF THE S
	tic [i-1] = i
	matures began
	algorithm: [P3 1 A 2 M 2 4 1 2 1]
	1 make a board and initialize the value
	1 Hake a winner function check for the
	winsul possibilities
	input.
	(i) imake a computer function:
	(i) Maximize computer wins
	(i) Maximuze user wins
	(ii) yourde valde position.
-	Light Call the Light Called the C
7072 - 80 73 - 60	as they can be a 3 combination ofter 5 steps (i) put man function
MU(を) 以上	prix the board
842	ad or and
A	ad shift byw the user fra puter.
- 1/1	SALK WIN
A	AMI .
1/1	Warra Salah - due
	(at 10 Margar 153 11) toget) to it states at
	The state of the s
	at the many and a state of
	The state of the s

[1, 2, 3,	4, 5, 6, 7	, 8, 9]	computer's	turn :		Your turn		
		+ !	+		 +	enter a nur	mber on the	e board :5
1	 2 	3 3	 0 +	 2 	3 3 	 0	 X	
 4	 5 		 x 	 5 		+	 0	
 7 	 8 		 7 	 x 		+ I I 7	 X 	
+ computer's +	turn :	+	Your turn enter a nu		 e board :3	computer's	turn :	+
1	 2 	 3 	+ 0	 2	 	 0 +	 X 	 0
 4 	 5 	 6 	+	 5	 6	 x 	 0 	 6
+ 7	 X 	+ 9	 7	 x	 + 	 x 	 x 	 9
four turn :		e board :1	 + computer's	 turn :	 -	Your turn enter a num		e board :9
 0	 2		+ 0	 X	 	 0 +	 X 	 0
 + 4	 5	 + 6	+	 5	 6	 x +	 0 	 6
 + 	 			 		 X	 	
7 +	X 	9 +	7 +	X 	9 +	+winner is	0	

	Page
2	frogram to solve me 8-puzzee using BFS.
[[(,ide	from collections import deque () double reded quene.
	dy find-blank (board, room to find zero (brank)
	for Ei in range (3):
	for j in range (3):
(,000)	board [it] == 0;
	1 650 com _ 19 2 1 may returni, 5 y
	initary (sourm, shope) chart for
((Tourne)	def generate mores (board): fuction to generally
	moves = []
	blank your, blank-col - find blank (board)
	possible_moves = [
	(1,0), (-1,0), (0,1), (0,-1)
	J Jan John Store July
(the second and the second
	of for dry de in possible moves:
	new row, new-col = blak-row+dr, blak-col
	out of new row (3 and OC= new-col/3:
	new board = [soup[:] for row inboard]
	new-board [black_low] [blank_col],
	pour == 1 copyratet newsboard (new-row] [new-col] = newsboard
ale be	[new row] [new board (blat 900) (blat -co
	moves append (new-board)
(010-	stote moues.
	A 15 and the MANUAL STATE TO HERE & CO.
BI	dy solve-pupple (indial-state, goal-state):
-	visited = set () State of their pathos
	queue = deque ([(intial - stock, []))]
	(* 12 August 19 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

	while queu!
	and the costs of path = quette. popular
	visited add (tuple (map (tuple, current-state))
-	Clares ()
la-c	if current state = = goal state:
	notine pate
	(A) who is a summer or in the
	possible moves = gurate moves (avount - state)
	for move in possible - moves
	if tuple (map (tuple, move)) not in visited
Lace	queue appeired ((more, path + (mon)
	TO TO A Second
	(desert short word of the short
	J = ARRIGINA AGRICORG
	dy puit steps (solution-path):
	if solution-path:
	penil (" Steps to reach him goal!")
	for Step in solution pate ?
الماعدات دماؤه	= + cust_ had - les - cumperal ("cust ")
	pleasing = 11 by a conforman in step:
أتعلم وعرط	Description ("1", ed = " ")
75.1	i crown lov ropaled [blank how of merces
Lead Cultis	her will toph thin broadwin of val == 0:
	on should become the committee and part ("" , ceda"
	(bunch-aux) Liego - Evenly !
	poit (Val, end = "
	penit ()
	(some with print print a street - was) for
6-474	prid () William
[((=	green - degret; Bright of make, (
	prid (" No solution")

ruge
intial = [
18 1 15 (1,2,3], [4,0,5], [4,7,2]] 15 11
goal = [(0,1,27,[3,4,57,[6,7,8]]
(1, 2, 6, 5, 5, 6, 7, 8)
Solution and
Solution pater - color-puggle. (nt oil, goal)
print steps (solution pate)
18/9/19/18/19/5/3 3 8/1/2 5, 4,5,6,78
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
13 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 2 3 1 5 1 1 2 8 dis up de did
18 1 2 6 7 8 1 2 10 will decusing a doubless
18 1 1 2 6 7 8 9 1 guene and the party and water
,
(0) using fuction
for i in rage 31
for jarages
y board (i) [j]==0:
b mari.
1411 John 3 18 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
(3) Nous ganside moves using function is none
the blankspace in all possible ways 14=
board sound (3 x3)
Obleto Ponce (5713)
(ii) if it is in bound generate the more
èc mover brake space with adjacent tiles
and board fa path.
use 66s now use visted list
every time generate a copy and check
with goal stale.
100

1112131 12131 213
1 1415 1 1 1 4 15 1 1 1 4 15 1
1617181 1617181
Colorina pate - color page Labor 2006)
2 3 5
1 (1 4 5 1 4 1 1 1 1 1 1 1 1
1617181 16 7 81 16 17181
[2[3][[2] [5] [2]5]
1617181 16 7181 617181
1 61 (10)
1) (at the black space during (2) us on truction
1112151 1112151 111215
1 3 4 31 4 13141
0 617 18 1 617 181 1617 181
130 -200
1 1 2 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 1 2 1
1514 J
The state of the s
X / S V S V S V S V S V S V S V S V S V S
- Mar
CON MONDER DELICON NI DE (11) 120
the moist bush spare, with
and board for party
Jail horest use the late
Asset has veres a let a come che che
the state of the s

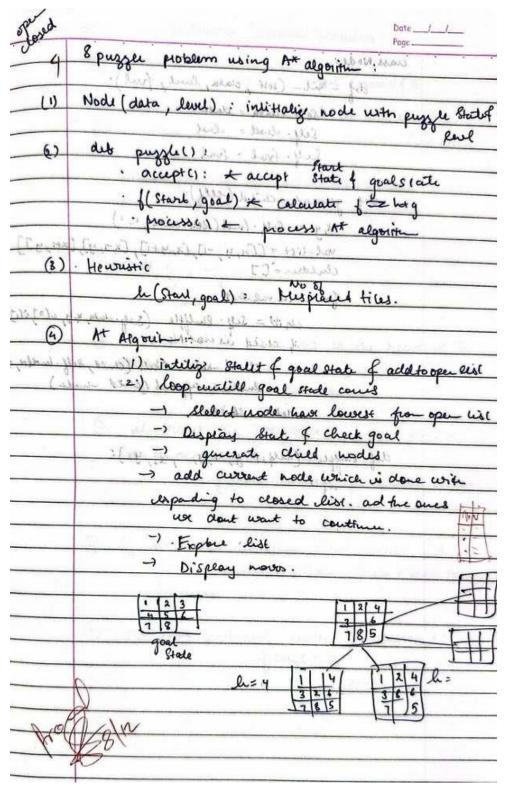
```
main.py
                      d=[]
                      if b not in [0,1,2]:
    d.append('u')
                     if b not in [6,7,8]:
    d.append('d')
if b not in [0,3,6]:
    d.append('1')
if b not in [2,5,8]:
    d.append('r')
  v 🗸 🌣 🔏
1 | 2 | 3
4 | 5 | 6
0 | 7 | 8
1 | 2 | 3
0 | 5 | 6
4 | 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 finished with exit code 0
Press ENTER to exit console.
```

8 puzzle iterative deepening search:-Code:-

	hab-5 Pege
8/12/2	the id- id- id- id- google, goals, god-oraness);
3	8 puzzle frobern owing therales deeping souch
141	algaritum. (1140 1140 1140 11
	algoritum. Carque Man 127 M
(b)	Node (data, level): initialize mode with puzzle sto
	flevel = Ci Takwer of
(2)	of puzze to home so
	- Start & goal Stat
pin Louisi	-> dls () (mode, goal, depre) - ferfor de
(3)	dy des (mode, goal, depter)
10	If (current State == goal):
	(Aget Calson) & Bretween goldion
Lipt	Z John Muser El
1216	llu:
269	generate child node, recursively call
	1) assessed - D = levith increase level
	in a makin (0), 2)1 police
Cw	dy 105 (Start, goal)
	Start with depth : 0
(2,0	Jane Marie Marie
Co.5	raise a sepect with your is found.
-	fuson Des with current depth
-(4	7 & Sola found = exit
	uncremed depth.
(6)	Alumode market structure of the structure of
(3)	-> yeurat puzzle distant
)	call ios (start, goal)
The	NW NAME OF THE PARTY OF THE PAR
MIL	A (course) have
1	(And " cold") Ind
-	NEW YORK TO THE TAXABLE PROPERTY OF THE PARTY OF THE PART

- 1	dy id-dis (puzzle, goal, get-moves):
	dy id-dis (puzzle, gode); import itertools
اعبدار	
	if depth == 01
15.14	VALALINA
2013 1	if south [-1] = good
	godin grante
	for move in get - moves (route [-1]):
	I was not in saul!
1.63	if next - quete:
all -	if next- goute:
	(wood choop abomoutal next 2 route (8)
	for oligh in identicals count ();
	route = dfs ([puzzle], depth)
_	if route:
	viela_ vioute
-	dy possible - moves (state): Output:-
Sec.	successif
2.7	of bushing (o), 231 poin:
	if 6 not in [6,7,8] [[1,2,3,0,4,6,7,5,8],
	a. append (' d') [1,2,3,4,0,6,7,5,8],
	af 6 mod in [0,3,6] [1,2,3,4,5,6,7,0,8],
M. C.	pas noors = [] [1,2,3,4,5,6,7,7,0]
. 1	1 12 12 1
	pos noves append (genrate (state, i, b))
	initial = (1,2,3,0,4,6,7,5,2)
	goal = [1,2,3,4,5,6,7,8,0]
	route = id - dfs (milial , goal , possible words)
	. If route:
	print (" Success")
	prix ("Patri: ", patri)
	puid (" Jarled to find Solution").

```
d.append('l')
if b not in [2, 5, 8]:
    d.append('r')
  30
             pos_moves = []
                 pos_moves.append(generate(state, i, b))
             return pos_moves
       def generate(state, m, b):
    temp = state.copy()
                 temp[b + 3], temp[b] = temp[b], temp[b + 3]
                  temp[b - 3], temp[b] = temp[b], temp[b - 3]
                m == '1':
  temp[b - 1], temp[b] = temp[b], temp[b - 1]
m == 'r':
             if m ==
                 temp[b + 1], temp[b] = temp[b], temp[b + 1]
             return temp
      # calling ID-DFS
initial = [1, 2, 3, 0, 4, 6, 7, 5, 8]
goal = [1, 2, 3, 4, 5, 6, 7, 8, 0]
       route = id_dfs(initial, goal, possible_moves)
            print("Success!! It is possible to solve 8 Puzzle problem")
print("Path:", route)
・ パ な 当
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 finished with exit code 0 ress ENTER to exit console.
```



	crass Node:
	dy - mil - Csorf, data, level, fral):
toll sto	all data = class
Seas or	Self elwel = clark
	Suy. Aval = Aval
الد	The state of the s
	del accurate - child ()
	ky = fill find (ser, o
	val-list = ((51, y, -1), (x, y+1), (x-1, y), (x+1, y
	val-list = ((31, y, -1], (x, y+1), (x-1, y), (x+1, y) childre = C]
	low in the val test on half
	chib = Self. Shelfle (Self. data, x, y, ilo)
	of chied is not Name!
	durd - Node (chies, self lus
	chedre append (child hade)
ا محالات	and resource of the shildren .
1	Toop to grant Start & charle goal
	dif suffic (eng, puz, x1, y, , 42, 42):
ه دید زمر	add contact hads that a day
2300	المهمالية أن ملافقية للأولاء ولما الله
	has along which to constitute
	- English Will
-	- A Distance property of
1	
-	
	1 600 June 12 V
= 2) 1	4 3 1 1 1 1 1 1
	Hait Hall Land

```
< 2 $ g
Enter the start state matrix
1 2 3
4 5 6
7 8
Enter the goal state matrix
1 2 3
4 5 6
78_
1 2 3
4 5 6
 7 8
1 2 3
4 5 6
7 _ 8
1 2 3
4 5 6
7 8 _
...Program finished with exit code 0
Press ENTER to exit console.
```

	Vacuum Cleanear Problem. Page
2	algorithm: define room 4 &B
	1 initial Chate - Command
	1 detions → Either in room A on B
	more L, move R, Suck (clea
	Success (1) 1150 (160)
(181	A. I machine reaches
	Successor function :) where he machine reaches
	Result (45) a) with strong
	parit (" Many as as as as
	state actional yes
	whither city class of the room its in
	to mext of de references () - tie)
	to next (with more function).
	(3) il runs the goal State Court ton atto anom
("). ~v	3 il remo the goal state function atta every
()	3 il rems the goal state function after every
(1. J. 20	3 il rems the goal state function after every
("}!~	3 il reus the goal state function after every if room A == clean ff room B == clear print (" cleaning complete"
	3 il resultative goal state function after every if room A == clean ff room B == clean print (" cleaning complete"
11/200	(3) il realis the goal state function after every if room A == clean ff room B == clear fruit (" clean g complete" (9) If that clean it simil such function and
(1).	3 il resultative goal state function after every if room A == clean ff room B == clean print (" cleaning complete"
([·	(9) if not clear it said such friction and returns clear of such friction and returns clear of such friction and returns clear of such friction and returns clear on their moves to
([:	(3) il result the goal state function after every If room A == clean ff room B == clean fruit (" cleaning complete" (9) If that clean it draws such function and pretures cleans then moves to
(")	(3) il realis the goal state function after every Step on action if room A == clean ff room B == clean print (" cleaning complete" (B) If that clean it downs such function and pretures clean then moves to the moves to
	(3) il realistic goal state function after every If room A == clean ff room B == clean print (" cleaning complete" (9) If that clean it raise such function and pretures cleans then moves to the have path cost counter. (each step 1) if count == 2
	9 If what clean it saws suck finction and returns cleans then moves to (5) We have path cost counter. (each step 1)
	(3) il realistic goal state function after every If room A == clean ff room B == clean print (" cleaning complete" (9) If that clean it raise such function and pretures cleans then moves to the have path cost counter. (each step 1) if count == 2

	1 dads:	thropho
	B) Amon suite.	Walter T.
	milial State - Singapagado	0
	A PA TO OCCUPANT	(£)
فريس	class Vaccum Cleaner:	
	class Vaccum Ceration.	υ <u>Σ</u>
salms	class vaccion testes (self) interest of the class of th	10/C/ [A/ 10/
.00	Ruf location = wandom . ch	march 10.
	dy move_lift (self):	(Q ()
	print (" Morring left") sey. location = (A)	6JP1
	self · location = (A)	
		5 = 6/2/ 1
p3 241	pruit (4 Moving seight"	Ju
News	in most for species consisting out	A Comment
	. (. in Self · location = 68'+ x sen	
	A. D. Campo C. Sall Proposition 1	(8)
وياري	of a med suck (self, room live in	0 - L 2'
	print (4" Sucking dirit in	Koan [sour]
	hower wiselien of clear?	
ولوان "	paid (cirain g com	
	def simulate_cleaning ():	
ا	of simulate _ cleaning ():	(2 B)
be)° (a)
be	vacuum = Vacuum Chann ()	i (dvily 1)
be	'A': vrandom · Choice ([Clean	
01-63-	'A': random Choice ([clean 'B': random Choice ([clean	· , · distry])
01-63-	'A': vrandom · Choice ([Clean 'B': vrandom · Choice ([Clean	· , · distry])
01-63-	A': vrandom · Choice ([incleand of the change of the chang	· , · distry])
01-63-	"A': vrandom · Choice ([i clean 'B': vrandom · Choice ([i clean 'B': vrandom · Choice ([i clean get and [i i i i i i i i i state!])	i astij]]
01-63-	print (f" Vacuum leaver is in A	oom Shame
91-63-	"A': vrandom · Choice ([i clean 'B': vrandom · Choice ([i clean 'B': vrandom · Choice ([i clean get and [i i i i i i i i i state!])	compression (1)

-	Page
	if sooms ['A'] == (clean and scores ['B'] == (clean'
	praise ("Both rooms are clean, No clean
	tule and a needed)
1	ulse house some product
	puil (4 C)
	current trading the cleaning process)
	avount - moon - va cum: location
	cleaned - room = vacuum suck (current - 900
	Vacanas cleaned in the Roma &
_	if cleaned -room = = ! clean!:
	crosmo [current & room] = "clear
	2 /
	ij wout - 200m == (A'
	Vaccum : many sight ()
	current-toon = 181
	ulse:
	Vacceum more left U
	current goon = (A)
	Agon = IT
	possit (" In (leaving completed . ")
	puit (t' Finial State")
	the final state ")
	puit (f" Room A: { rooms ("A)]}")
	penil (f" Room 8: f. rooms ["B']")
	Simulaty_cleaning()
	J. (
	output:
	. /
	Enter initial location of vacuum cleaner (A18): A
	Enter State for Room A (clean/sturly): divity
	Ender State for Room B (clear / durly): dirly
	Indial State:
	Vacuum Cerany is in Room A
	Room A: diray
	Room B: Clirty

	Sucking the creaty process
د اردوس ،	Starting The clean R'o on A
to clearing	
(,	suchig die in Rooms
(* 14)	and principle and principle of the control
	clear g completed
(moor trees	Foral Stale:
	Vacuus cleams as
1 1	Room B : Clean Box Clean
Nun34	(to) == moore - tween in
£.,	(4) == maore - pro-
	A) type moon may to
_	181 =
	(1 that some from consumption)
	1911 comment therefore
	point (" you Charing completed . ")
	poit (f. Ream A: f. moons (19.74")
	(* (* * * * * * * * * * * * * * * * *
	Emulaly clean 3 ()
	The second secon
	Sulpert:

Output:-

... Program finished with exit code 0

Press ENTER to exit console.

```
main.py
              nonlocal cost
              if goal_state[room] == 1:
                 print(f"Cleaning Room {room}...")
                  goal_state[room] = 0
                  cost += 1 # Cost for cleaning
                  print(f"Room {room} has been cleaned. Current cost: {cost}")
                  print(f"Room {room} is already clean.")
          # Cleaning Logic
          rooms = ['A', 'B', 'C', 'D']
          current_index = rooms.index(location_input)
          # Clean all rooms starting from the initial location
          for i in range(current_index, len(rooms)):
              clean_room(rooms[i])
          for i in range(0, current_index):
              clean_room(rooms[i])
                                                                                      input
Enter Initial Location of Vacuum (A/B/C/D): B
Enter status of each room (1 - dirty, 0 - clean):
Status of Room A: 1
Status of Room B: 0
Status of Room C: 1
Status of Room D: 1
Initial Location Condition: {'A': 1, 'B': 0, 'C': 1, 'D': 1}
Room B is already clean.
Cleaning Room C...
Room C has been cleaned. Current cost: 1
Cleaning Room D...
Room D has been cleaned. Current cost: 2
Cleaning Room A...
Room A has been cleaned. Current cost: 3
Final State of Rooms: {'A': 0, 'B': 0, 'C': 0, 'D': 0}
Performance Measurement (Total Cost): 7
```

Knowledge-based entailment:-Code:-

	in / / P-1 11 2000 1
0	Entailment gall Poge and if Poge
16	Franks:
	1 Charles appropriate for
	Inputs:-
-	
	knowledge have (
(4)	De logical rules
	facily statement!
	knowledge base (set of logical rules)
	Crops:
	Ereps:
	1.) Negote the query : Obtain the negotion.
	and substitute of the superior
((parasaget)	Obtain the negation.
	2.) Combine with kings that
	2.) Combine with knowledge base.
	3.) Check Cats Cincal
	3.) Check Sates fiability: to check if the negation with kb
	to check if the negation with ks
	res Satisfing therules
	(patering (2)) Strapes parties at a soul
-	4.) Determine entailment
2 week	ist committee in the same of
pried	I sugarist as not salisfied > Irue
1	if conjunction is solistiable - thre.
-	
	- Wangkasi
2(1) 644	milly (M. N. charge +) I may a madage between y
	great agentique de la company
	- Reserve to the second of the second
	

	The state of the s
	From sympy import symbols
A Park	from sympy impore sympose
	tel matte kunting
	p = symbols ('p')
	= symbol ('gy')
	y = symbols ('q')
	V = Symbols (1)
	peraverage base = And (implies ())
	V = Symbols ('(')) V = Symbols ('(')) Movedge base = And (implies e(p, ov), implies (v,)) Not (r)
	relan knowledge base.
	def query - estails (knowledgebase, query) entailmeis = salufiable (And (knowledgebase, Not (query)
	edulicity (And (knowledgebase,
	Not (query
	rete not entailment son son son
	in- name_ = "
1254	Kb = create tenowedge base ()
	query syrbols ('P')
	resultiff = query-entails (Kb, query)
	prit (" knowledge Base ", Kb)
1	
٠.5	pril (" Query entails bowledge base", result
	Output:
	Knowledgebase: ~ & (inpluis (P, W) & (Inplus (V)) Query: P
	Bury: P
	0
	away entails knowledgebase, False.
-	

```
main.py
          entailment = satisfiable(And(knowledge base, Not(query)))
          # If there is no satisfying assignment, then the query is entailed
          return not entailment
  25 if __name__ == "__main__":
         # Create the knowledge base
          kb = create_knowledge_base()
         # Define a query
          query = symbols('p')
          result = query_entails(kb, query)
         print("Knowledge Base:", kb)
          print("Query:", query)
          print("Query entails Knowledge Base:", result)
v 2' $ 9
Knowledge Base: ~r & (Implies(p, q)) & (Implies(q, r))
Query entails Knowledge Base: False
...Program finished with exit code 0
Press ENTER to exit console.
```

Knowledge-based resolution Code:-

-7	Knowledge based resolution!	Maple in Con
	1 12 of tall bart	d. No.
	Inputs:	
NT COLUMN		
	Knowledge	1 IV 5 N
	1 class (set of class	uses in propositional
	Knowledge base (set of class	logic)
	Steps:	Latter Bully
60	MATERIAL PROPERTY	
	1) Frilancia relative 1.	\
	1) Frilialize vesselvent:	v 1/1 - 1 - 1
	(2) Repeat Gutill no new ires	bourts can be guesaled
10	20 Paining will	1 th ries
da) x	Colonse	A , clause B)
topad	of total a deco	
1816	3:) Resolving clause.	
/		
Libril	shirehous) again waters	
	BVA	P,
	- CVA	TRUG
	¬ B∨A	PV-GVR
	CV70, 2 - ALLIAN LINE	/ \
	- AVTBUD	7Q VR
-		615100
	A - Roud	(1/2
/	Course States and	
1	Megate the avery fadd it +	0 tu kb
1.1.	0	
7-253.	- Repeatedy tresolve the pa	D
in'uni	knowledge last untill	a contradiction found
-	No men sules are pos	will.
	Marcel	,
	ARI	
	AND THE REAL PROPERTY OF THE PARTY OF THE PA	Parameter
	the state of the s	After the second
	- April - Apri	

	dy negatie-libral (tileras) sif libral [0] == 'n': sif libral [1:]
	dy negatie - state (n' :
	sil literal Los
	sif libral [0] = 1: J
	else: ret 'n' + televal
1 - p. 2V	many of the contract of the co
-	(3000)
	d. 5.10. (1. (2):
	dy resolve (1, C2): gusdue-clause = set (C1) Sut(C2)
	gusolve-clause
	for elebral in Collection
ha kesa	for eleteral in College (Ideal) into:
	inegati state
	resolved-clause remore (w
	gestly clause - levour my
	literal (lite
	and I I I wanted that
	oreten laple (resolved-clause
	A LAW ESTA TOTAL
	dil resolution (knowledge base):
	dig resolution (knowledge base):
- 1	write true:
	new clauses = Set ()
A.	for i, c, in during ale (Kb)
	of hear ! zz; A.
	new clause - resolus (Grass
	heis clause - resour (or
	if le (new class)=0
	brown of house all interest of his good and
n hand	new-Chure add (wwill
	of next was a second
	not new-clause
	kreak
	Knowledgelood
	Knowledge bas - new -claus
-	Selen proudedge base
	- home = = " !
No. of the last of	Ma-

	kb = {('p', 'a'), ('np', (x'), ('ng', '~
	result = resolute (166)
	t (4 minual tet)" (ch)
-	put (" Preduct kis", remet)
4	Sealifern of a country to a part of
	Janitorbi ARD & most or house (a last identical
	And a definition of the second
	all Elist in Cham 1 is a retrieble
	Small secure in the secure
	AND DEAD OF AN ADMINISTRATION OF THE PARTY O
	to aline course & ready ju
	Capital - aspertants
	[(2-1) 1-22)] - median ((22-2))
	TALL SALE (No. 1)
6500	Company (Amounts) Amountained (Amounts) of properties of
	107
	Megas manager to remained at 1991
	Long Jones Con.
	atom Committee Statement

Output:-

	2
- 8	Unification
	Eg knows (John, x) Knows (John, Jane)
	Call of x 1 Tame }
	() C / Jame }
	Step 1: If term 1 or term 2 ils avallable as
	constant there:
	a) term 1 or term 2 are identical
	3 Seturn NIL
	b.) Else if team 1 is a variable
	if term 1 occurs in term 2
	ocetien FAIL
	() else if term 2 is a variable
	ij Lerm 2 oceure in term!
	vietne FAIL
	ulse
	d.) else reliens FAII
	d:) else trelien FAIL
	that they are the
	Step 2: if predicate (term!) of predicate (terms
	Section FAIL
	84002:
	Step3: number of degeneral of
	Stellith FAIL
	Stepy: set (sur st) to NIL
	Step 51 For 1=1 do the hunder of celements in
	Win 1
	a) call unity (its term, its te 2)
	ped tresulto vintos
	S = FAIL Sutis FAIL.

,	Steppes: c) wy S fril
	THE WAY DESIGN
	a.) Apply s to the overnainder & both 1, ELZ
	6.) SUBST - APPEND (S, SUBST)
	SUSST APPEND (S, SURST)
	Step 6: Reline SUBST
	1 pudicate same
	6 No Z argunuts
-	(3) The 2011
	Jane 1 Association of the State of the Jane
1	((agra) tel signification of a subset
LaV	A STATE OF THE STA
1.	A contract of the contract of
	import ore
	def get Initial Predicate (expression):
-	echien expression split ("0")[0]
(-	def exconstand (chan).
	Delever chan Supper() ad Im(dian) == 1
	. def exeptace Attributes (very, old, new):
	attributes = getAltribules (exp)
	for index, val in enumerate (allributes).
	(194) 1-9 ugm val = = old:
	allei butes [index] = new
	predicate = getInitial predicate (exp)
1	Delver producate +" ("+ ")" · join (ettribute)+")"
-	def apply (oxp; sulsititutions).
	for Sulstitulea in Sulstitutions:
	New, ald = Sulviticulian
	exp - replace Attributes (exp, old, new)
	eletten exp.
	- Curation of the control of the con
	def getfvistPart (expression):
	predicate = gottnitial Predicate (asypunian)
	attribute author of the species
	attributes = getAthibudu(expression)
	irelien new Expression.

	1
	defunty (exp1, exp2)
المدارية و	
	Terror Vertical
	if its constant (exp 1).
	retur [(exp2, wq1)]
and the fallows.	i vivocitor (exp1):
74	of check Ocurs (exp, exp2):
	return Fall
destribution of	attribute (exp.)
وسلا أاستعلقيه	albubile Count 2 = len (get Attibules (exp2))
	of allribut (Count 1 = attribute count 2:
	ereture talse.
	1 - a.d. Fire liftenet (ox a)
	head = get firstpart (expl)
	head 2 = get tiret Part (pop2)
	wital Substitution - unify (heads, heads)
	if not initial. Bulestitution:
	Can delta False.
	Ocale in the land in the land
- (60	tail i = getReman and (cexps)
	tail 1 = getRemantant (cexp1)
	tail 2 = get RemacigPart (exps)
	Bredigate - John Tales - Antidate Com
" Carpin	into al Substitution , extend (oran a Substitution)
	sules ind al Sulstitution.
	Liministra de manajeros estados de la constitución de la constitución de la constitución de la constitución de
	man francisco de la company
1 Carre	No. 1425) Islandie Wichenspag - 123
	Distance Copies
	the get from the all expressions.
(***	the greekinds - gottining Abrechinds Curpus;
	atesibuly = goldenia dulus grasias)
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```
exp1 = "knows(X)"
exp2 = "knows(Richard)"
substitutions = unify(exp1, exp2)
print("Substitutions:")
print(substitutions)

Substitutions:
[('X', 'Richard')]
```

```
[7] exp1 = "knows(A,x)"
    exp2 = "knows(y,mother(y))"
    substitutions = unify(exp1, exp2)
    print("Substitutions:")
    print(substitutions)

Substitutions:
    [('A', 'y'), ('mother(y)', 'x')]
```

DOT A TOP	FOL to CNF coursing Page.
9	Stepi) create a list of super contract
	Steps) count a list of SKOLEH. CONSTANTS Step2 Find 43 -> pure points
	4 forms
Sandaria .	of the allering of
	both a could are lower case, suplace the
	sperim constant.
	remove resid spotem constant or function from
	The state of the s
_	if the allribuly are bothe lown case and
-	appearable Suplece the appearance albeitutes w
-	a Skolon fundion.
_	Warner (weeken) had him to be a second
	Stop3 : replace & with '-'
	braneform - as Q = (P ⇒ Q) ∧ (D ⇒ P)
	(Arr. The 1) should be seen to proper the
	Stopy: replace = with (-)
Section 19	doubling appropriate and a second
	Stops: Apply de morgans law
	· Complane ~ C
	: as ~ Pf ~P ob (1 was present)
	suplace ~[
	as approp if (f was great)
	supeace in a wir " "
	TO POOR SPACES
	Control of the Contro
	+ 1 King (x) A lexuedy (x) => Evil (x)
	The following (E) - Stril (E)
	~ [King (R) A greedy (R)] Vail (R
	(King (E) / grung (E))
10	~ (K = (x)) V ~ (eneedy (x)) V Ci
tro	(R)
L. Wales	galleting the sale of the sale of

dy getAthibutus (Stung!): expr = '()] + mothus = no. Andall (chippy, string) return (m for m in str (mothus) up on isalphan) dy got Boudicalis (throp): expr = '(a-2 ~) + (A-Zr-z] + ' expr = '(a-2 ~) + (A-Zr-z] + ' exetter on Andall (chippy, spring) def OreMorgo (lenterie): string = " join (list (senturus) · capy ()) String = String · replace ('~",''') frag = '(' in String) for predicate in get Dicticate (string): gray = string · repeace (predicate, f'-forme S = list (string) for i, c in enumerate (string): elif (== '4': e
modelies: no. fundall (elips, string) return (nom for one in str. (modelies) up on isalphane) dif get soudicable (string): expr = '[a-2~]+(A-Za-z]+' return on foundall (elips, string) def De Morgo (lentence): string = ". join (list (sentence): string = String replace ('nom', ") frag = '[' in string string = string repease ('nom', ") for prevaicate in get succicates (thing): string = String repease (predicate, thing): string = String repease (predicate, thing): string = String repease (string): string = String repease (string): string = String repease (string): string = 'string repease (string):
return (m for m in the (matched) of m. isalphan) dy got Pondicadle (large): expr = '[a-2~]+(A-Za-z]+' verturn ere. findall (corps, string) dy Dre Morga (lentence): string = ". join (list (sentence) · capy ()) String = String · replace ('~~','') frag = '[' in string string = string replace ('~['') for precision in get Discicatu(stlig): string = string repeace (predicade, f'-formed S = list (string) for i, c in eneminate (string): string = "!': string = "!': string = "!':
expy = '[a-2~]+(A-Za-z]+' vette en Andall (corps, string) def Die Morge (sentence): string = ". join (list (sentence) · copy ()) String = String · repeace ('~~','') for predicate in get Discicate (string): string = string · repeace (predicate, t'-[onit S = list (string) for i, c in enumerate (string): string = 'f' string - string · repeace (predicate, t'-[onit S = list (string) for i, c in enumerate (string): string = 'f' string = 'f' string - string · repeace (predicate, t'-[onit string = 'f' string - string · repeace (string): string - stri
expy = '[a-2~]+(A-Za-z]+' veether ere fondall (corps, string) def Die Morge (sentence): string = ". join (list(sentence) · capy()) String = String · repeace ('~~','') frag = '[' in String string = string · repeace ('~['') for predicted in get Discitate(string): string = string · repeace (predicted, t'-[onite S = list (string) for i, c in enumerate (string): string = 'f' selef C== 'f':
def Dre Morgn (sentence): String = ". join (list (sentence) · copy ()) String = String · replace ('~~', "') frag = '[' in String String = string · replace ('~[', ''') for predicate in get predicate (string): String - String · repeace (predicate, f'-[mids S = list (string) for i, c in enumerate (string): **String = 'g' velif C = '1':
def De Morga (lentence): String = ". join (list (sentence) · copy ()) String = String · replace ('~~','') frag = '[' in String string = string · replace ('~[''') for predicate in get Predicate (thing): String = String · repeace (predicate, t'-[mids S = list (String) for i, c in enumerate (string): elef C = '[': slif = '[':
String = ". join (list (sentence) · copy ()) String = String · replace ('~~','') frag = '[' ii string string = string · replace ('~[', ''') for precisions in get predicate (string): String = string · repeace (predicate, t'-[mids S = list (string) for i, c in enumerate (string): **S[i] = 'f' cely C = 'f':
fing = '[' ii string string = string .replace ('~[', ''') for predicate in get Predicate(string): string = string .repeace (predicate, t'-[print) S = list (string) for i, c in enumerate (string): if C == '1': slid = 'f' cely C == '4':
for predicate in get Predicate (this): Stry - strig repease (predicate, t'- [print) S = list (strig) for i, cin enumerate (strig): "If C = '1': **EliJ = '5' **Cely C = '4':
for predicate in get Predicate(Alig): Stry - Stry repease (predicate, t'-family S = list (Stry) for i, c in encourate (Stry): If C == '1': S[i] = 'f':
S= list (stig) for i, ciù enemerate (stig): S[i] = 'f' cely C=='4':
S = list (stig) for i, c in enumerate (stig): $ \begin{array}{cccccccccccccccccccccccccccccccccc$
(1) (1) (1) (1) (1) (1) (1) (1) (1) (1)
(Learne 10 0) (E = 11: 11: 11: 11: 11: 11: 11: 11: 11: 1
(Lung (La Co)) = (S (i] = '\$' :
(hand sour) Jewill CH = 141:
Sty = ", jan (s) Sty = ", jan (s)
Sty : Hy replace ("~~", ")
Sty : Hy replace ("~~", ")
Clare to the contract of the c
viete T (Stig 4) wiften sends
() Soil () () property of () ()
dy skolemization (senture):
SKOLEM_CONSTANTS = [f'fchn(c)] forc in reg
(ord ('A'), ord ('z')t
Stalent = " . jai (lest (senture) . copy 4)
materia = re. findale (1/1+37. , statud)
for matches in re matour [::-1]
The state of the s

,	
	for predicate in get predicates (Hote)
	attributes - get Attribute (predicate)
THE SAME	join Callettules) is cours (1:
	Statement : Statement replace (matchle),
	skolen cortails : pop(a)
	else:
	al = [a for a in all ribert if a iscours]
	all = [a for a in albubils if not a recond
	Co7
	Julian State and
	1 Million of the Car
	while '-' in Statement:
Ab Lotte	Gratement Andex ('ny')
	Statement - lest (state and)
	Statement [i] = , Stalent [iti], Stalend(itz]="7
	Statud (i+2], (~)
	Galement = 5 's jain (Statement)
	Statement : statement replace ('N[+', '[~+')
	((E +)a))'
	for Six Stabilis
	Cops = in!
/	for S in Statements:
	reta Statuil.
	Julia State
	and the state of t

```
print(Skolemization(fol_to_cnf("animal(y)<=>loves(x,y)")))
print(Skolemization(fol_to_cnf("vx[vy[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)"))

[~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)
```

10/1/20	Page 1
10	Forward chaining
	1) Input the knowledge close and the query
14.670.00	2') for i in KB:
	n (+)
Denistra Denistra	ceptur your act in part
	add sens to KB
	3.) To remove variables
	if 1. louses ()?
	ecepeace the varible with constants
3-5-53	Example: 1 And 1 - [3] And 100
	KR (2) & greedy (c) => evil (2)
- Am) 1	lag (John)
	greidy (John) Keg (Richard)
	Query : Le mary ni ?
15 000	A STATE OF THE STA
	The state of the s

_	Code: Page Page
-	Simport due (v) 10 march 1 1 march 20
_	del is variable (x)
	Jacket llu(x)== 1 ad x islamen () ad x is alp
_	ad x. is alp
_	of get Attributes (Phy)
_	esper: ')([^]]+1)'
eh.	return matches.
1	(CAS) see Ask
	def get predicates (8tmg):
_	Ocpt = ((a-z~)+) // [^ 4[]+1]
-	vielen re. findall (expr. strig)
	class Fact:
	def inie- (self - expression)
han	self expression - expression.
Jai	self. parais: parais
S.	Self · result = any (sex getconstarts (1)
	duf split Expression (self, cexpression):
	predicate = get predicate (expression)(0)
_	params: gt Atributs (aprusia (0))
	(p (()) . split (',1)
	Aulum (predicate, pares)
_	del geteresul (ee4):
4	geleen leff result.
1	de get Constant (enf):
114	Holyen None if it voulte (c) west of
-	an self parus]

	-
ay get variable (self):	
scelver (v if is variable (v) alse None of	er Vii
self parus]	
Court of the Control of the second of the se	
Class Inplications (Buy expression)	
self. legressier - expression	
(= 13 (=) (= curere in specif (=))	3
self les = fact (1) for unite]. speid
suy sus = fact (1(1))	
def evaluate (self. facts):	Sanc .
(and ages) constants = { Zulan.	
Nur Chr-67	
for fact in facts it was	
for val in left lls if val-predicate = fact	4 144
(+ tal-producate-fact	(val.gody
(1) decreasing the property of leverest	arially
	0)
L class to: () is surged talan Jul	
(suy):	
self facts set ()	
Rely . Implications = Set ()	-
des tell (sey-c)	
in c:	
Self-inplications. add	(Xplicaling
for i in seef implicat	
12 12 12 12 12 12 12 12 12 12 12 12 12 1	y.fades
Fanker 1952 -45	

Oute
dy gury (sey . 0): facts: set ([+. " pussion for in sey facts] dy despeay (self):
facts - see (5
(f. 1x press on for it
def despeay (sulf): period ("All facts")
period (" All
(facts")
() the last of the
for in sey facts print (fee fix ? [53]
print (fee Six 12 1 co)
146 = KB()
the - tell ("Knig (x) of greedy (x) =) evil (n))
to - tell ('King (John)') predy (x) =) evil (x))
to lell (greedy (Tol.))
Kb lill ('gruedy (John)') Kb - dill ('Kij (Richard)')
to gury ('evil (x)')
Output:
Georgie (vil (x):
lvil (John)
ΔOV
24 1 2 9
A it.
Const

Output:-

```
kb_ = KB()
kb_.tell('king(x)&greedy(x)=>evil(x)')
kb_.tell('king(John)')
kb_.tell('greedy(John)')
kb_.tell('king(Richard)')
kb_.query('evil(x)')

Querying evil(x):
    1. evil(John)
```