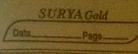
		/ Date Page	
	8pv33le problem	IBMI8CSOL9	
		Mk Gragom Roshan	
	import copy, math		
angle and the second	# Depth birst search in search of		
	target-using recursion.		
	target - [[1,2,3],		
	[4,5,6]		
	[7,8,0]]		
	def solve (sx, target, limit, visited states)		
	il sac - = terget:		
	print ("Target is knowled in sta	igz (limit)	
	roturn True		-
	If limit >= 12:		-
	return False		2
	visited_states.append(sac)	<u></u>	
	actions - possible - moves (STC,	visited states)	
	new move=[]		3
	min = mathing		
	for action in actions =		-
	if action not in visited states	, and monhattan	
	(action, target, limit) = min:		-
	min = manhatton (cution	target limit)	-
	new_move =action		-
	print ("Stage:", limit +)		ŀ
	diglay (new-move)		_
	il slyg(rew_move, target, s	limit 1 / Visited states)	
	is Tare:	Ţ.	_
	Zoturn Falle		
	The state of the s		



dof Index (raylist, v): for i, x in enumerate (mylist). il u In x : acturn (i, x. Ender (v)) deb possible moves (state, vicited states): b - index(state, o) d=[7 if \$ 6 29. do appoint ('d') 4 6(0)-0: de append ('u') if b[17=9: de append ('n') if \$ 17 > 0: d. append (41) pos\_moves = [7 for i ind: mare = gen (state, i, b) if move not in virited states = pos\_moves.append(move) del gen (state, m, 5): temp = copy. dee popy (state) if m = -(u): temp[6[0][6[1]=temp[6[0]-1][6[1]) elif m - 'd': temp[b[o]](b[i])=temp[s[o]+)(b[i]) temp[b[o]+][b[i]]=0 elig m == "(": temp[b[][b[i]]=temp[b[0][b[i]-i] temp[b[o][b[j-i]o elif m= (3): temp[b(o)[Li)-temp[b(o)(Ki)+1 if ex== "Narget" return temp temp[6[0][6[1]+]=0