	Vacuum Cleaniai Problem. Page.
5	algorithm: difine room 4 FB
-	
	1 Actions → Either in room A of B
	(2) Actions - more L, move R, Suck (de
181	distriction withere the machine reache
,	Successor function is performed.
	(1)(2)(1)
	Result of S. a. S.
	and (Handy story
	Result "(45) a 2 man pool of the cational . Just
	tight first at checks in the moon it is
	1.64.4
	which cits clean or not if clean it more
	to next (dirtimment function).
	to next (distingue function).
7.0	(3) it rand the goal state function after every
	3 il rans the goal state function after every
	3 il rans the goal state function after every
	3 il rans the goal state function after every
	3 il rans the goal state function after every if room A == clean ff room B == clean paid (" clean g complete
٠٠٠٠ الم.٠٠	3 il rans the goal state function after every if room A == clean ff room B == clea print (" clean g complete 9 if not clean it divis suck function and
my my	3 il rans the goal state function after every
(L.	3 il rans the goal state function after every if room A == clean ff room B == clea paid (" clean g complete 9 if not clean it saws suck function and returns cleans then moves to
	3 il rand the goal state function after every if room A == clean ff room B == clea print (" clean g complete G if not clean it divis sutt function and returns clean then moves to
; [];	3 il rand the goal state function after every if room A == clean ff room B == clea print (" clean g complete G if not clean it divis sutk function and returns clean then moves to
	3 il rand the goal state function after every if room A == clean ff room B == clea print (" clean g complete G if not clean it divis sutt function and returns clean then moves to
	3 il rand the goal state function after every if room A == clean ff room B == clea print (" clean g complete G if not clean it divis sutk function and returns clean then moves to
	3 il rand the goal state function after every if room A == clean ff room B == clea print (" clean g complete 9 if not clean it divis suck function and returns clean other moves to the west of counter (each step 1) if count == 2
(L),	3 il rand the goal state function after every if room A == clean ff room B == clea print (" clean g complete 9 if not clean it divis suck function and returns clean other moves to the west of counter (each step 1) if count == 2

	code:
	The same same same same same same same sam
	Dankonadomis - del loiline D
	1000
e Lam.	ment / ment P for
	class Vaccum Cloaner:
Lielani	class vacción control (self) internet choice (['A', 'A
- Cara	Buy location = viandem choice (['A', 'E
	puit (" Moving left") Luder 1 self . location = (A)
	point (" Moving left)
	self · location = (A)
	and the second of the second of
	pruit (4 Moving right")
pi Zhi	1 (4 Marguella gright")
ANIMANA	in most for spents (arresting out)
	(waste self . encation = 9
	the of ay suck (say, room him is 6
E college	ATT D WITH CITY DICENTER PARTY TO THE PARTY
- 0	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	· pount (1" Sucking dirt in Koon 2 soom
	· pount (1" Sucking dirt in Koon 2 soom
دع بلازمین	print (t" Sucking dirt in Koon 2 soon
دع بلازمین	print (t" Sucking dirt in Koon 2 soon
عع طروس	print (t" Sucking dirt in Koon 2 soon
es dean	dy simulate_cleaning (): Vacuum = Vacuum Cleanin ()
es dean	dy simulate_cleaning (): Nacuum = Vacuum Channel)
es dean	dy simulate_cleaning (): No a course Vacuo Change (): A': vrandom · Chaice ([sclean', 'duity'])
22 steam plate " 	dy simulate cleaning (): vacuum = Vacuum Cleaning () 'A': vrandom · Choice ([clean', 'duity']) 'B': vrandom · Choice ([clean', 'duity'])
22 steam plate " 	dy simulate cleaning (): "A': vrandom: Choice (['clean', 'duity']) "B': vrandom: Choice (['clean', 'duity'])
22 steam plate " 	reserve de l'anne de l'ann
22 steam plate " 	dy simulate cleaning (): A course Vacuum Change (): (A': vrandom Chaice ([i clean', 'duity']) (B': vrandom Chaice ([i clean', 'duity']) (B': vrandom Chaice ([i clean', 'duity']) (B': vrandom Chaice ([i clean', 'duity'])
22 steam plate " 	print (f" Sucking dirt mekosn 2 soom a steeling soom clear? dy simulate _ cleaning (): "A': vrandom · Choice (["clean', 'dvily']) "B': vrandom · Choice (["clean', 'dvily'])
22 steam plate " 	dy simulate cleaning (): A course Vacuum Change (): (A': vrandom Chaice ([i clean', 'duity']) (B': vrandom Chaice ([i clean', 'duity']) (B': vrandom Chaice ([i clean', 'duity']) (B': vrandom Chaice ([i clean', 'duity'])

- T	Page
-	if sooms ['A'] == "clean" and scores ['B'] == "clean",
	petit ("Both rooms are clean, No cleaning
	tule's a suggest the needed)
1	- ulse done die produced
	puil (4 charies 4
	arount - room + va cum. location
	cleaned - room = vacuum suck (current room
	sack (coverent - 400m
	Vacanus cleaned is the Man o
	if cleaned -room == ! clean!:
	rooms [current from] = 'Clark
	: Current
	if wout - soom == (A'
	Vaccum · morey - right ()
	current-400n = 181
-	ulse:
	Vacceum mover left ()
	Current room - (A)
-	
	possit (" In Cleaning completed . ")
	puit (f" Fruial State")
	puit (f" Room A: { rooms ('A:] }")
	penil (f" Room 8: (.rooms ["8,3")
	Simulaty_clean=q()
	output:
	She finding the state of the st
	Enter indial location of vacuum cleaner (A/B): A
	Enter State for Room A (clean/courty): divity
	Ender State for Room B (clear journey): dirly
	Inclial State:
	a com a di Room A
	Room A: dirty
	Room B: dirty

	Stading the cleary process
ا درادسا،	Sucking dirt in Room A
needed)	
(Australia	mooning in Rooms
	n in head of
1	we clear g completed man - hamels
	() Foral That it : moor - benear
(moor town)	Vacuum cleaned is in Room B
	: Room A: Cleans - James 1
len's	Room & s. clean bourges
*:	in the == moore - through in
	Vacion moon myst (
	181 = wast - tweeton
	ردگارد :
	Vicenson, on the sett of
	151 - many - Ladden
	penit (" maining completed . ")
	point (Final State ")
	pait (In Room A: { noons(1.8.7) }")
	(* 5 . 8 . J zmans) : 8 moos . 4) x . 9
	Emusely cleaning ()
	The state of the s
	Gertpent:

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.")
          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
```