("my chuyam) Vacuum cleaner powhler aly-innit_(au) - (bacconoma obuymas top : dey, state & me for A or B ") = 190/6 vacuum-proj = infrus ('Enser A Ox B") room_A = input ("Enter obling of duity") and move egy () suppose for I make please in the second of the mind. : (B') self stark (nacrim-has) == (, 4,000) to vite work dy more right() { aj (all - state (vacuum-pos) = : A) dd gnak () é in few (sound): if (delf-state (vacuum-pos) == 'A')

if (delf-state (vacuum-pos) == 'A')

delf-state (vacuum-pos) == 'A')

delf-state (vacuum-pos) == 'A')) toward in 10 8, FL,) THIS else if (acy stare (vacuum pos): 6') ef (self state (room B) : duity") Ley day (room. B): "clean def main () ig lace state (vacuum. post:: 'A')
ig (acy state (vacuum.post:: 'A') du suck()
moveright() elving (self state (vacuum. por):: 'B')

y (self state (vacuum. por):: 'B')

suck () elxeanck() move. WHI) cere up (room. of: "dean" as soom. B: "clean") break

class vacuum Cleans ? del sun caul). def_init_(acef): i dubile not acci. in decort acy. state: { "vacuum pos : input (Enter the irritial position of the vacuum cleaner (AO(B): "). upper (), "scoom-A": input ("Is Room A durly or clean?"). Cour() "room-B": input ("Is Room B ducty or clean?"). Cover () ou ahow-state (sue): frunt (f baum lesition: {auf at alle fivacuum pos)}, Room B: { sey. state ('room-A')}

Room B: { sey. state ('room-B')}

dy-us clean (suf): retwen deef. state ["room-A"] == clean and self. state ("room. B") == "plean)" toto - welle you def move-right Coeff): () is in subject in musico uj sey atare ("vocuum pos"):="A": Onur invuent dell. State ("vacuum pur") = = "B".

prient l'"Moving la Room B") dy move-left (suf):

uj dey. state (vacuum. fros) == 18". self. state l'vacuum-post frank ("moving to Room A")

auch (sey). ig oul state ("vacuum. pos") = = "A". if deef. atale ("room-A") == "ducty":
self. atale ("room-A") = "clean" fruit ("Cleaning room 6")

defrun (sey): while not deep us wear (): deef show- stare () ig dolf. atout ("vacuum-pos")== "A". ing sep state ("100m-1")=="duty": dceg-auck() deg. move-right() elil dey star ("vacum-pos")=="B": up acy state ["room-B"] = "alurty ? dey. auch () &cep. move-lept() fruit (Both rooms are clean now acy. dhow-state() Vacuum: Vacuum Cleaner () vacuum.run () author : Enter initial position of the vacuum cleaner (Aor B): A Is Goom A durity or clean? clean Is Room B dury or clean? any Vacuum Britism: A, Room A: clean, Room B: dury moving to Room B Vacuum Rosition: B Room A: clean, Room B: dury Cleaning Room B both rooms are clean now Jacuum porition: B, ROOMA: Clean. Room B: clean

authul for four cooms:
Step? Vacuum is in room A, Room state: elean Action: move down moved down to Room C Room states: S'A': 'clean', 'B': 'Clean', 'C': 'Durty," 'D': Durty }
step 2 vacuum is in room C, Room State: Ruity Action: duck Suckeng duit in room C poon states: ('A: 'clean', 'B': 'clean', 'c: 'clean', 'D': 'Ducky')
Step 3: Nacuum is in room Cir Room State: Clean Aution: move signt moved right to soom D Room States: & A': 'Clean', 'B': 'Clean, 'C': 'Clean; 'O': 'Cuiry'}
drep 4:
Nacuum is in room D, Room state: Brity Action: Suck ducking duit in room O Room states: A ': 'clean', 'B': 'clean', 'C': 'clean', 'b': clean', self. room: { alf. neighbors = { 'A: 'clean', 'A: 'right': B', 'down': 'C': 'B': 'clean', 'B': 'Luft': A', 'down': 'D'; 'C': 'Quity', 'C': { 'uft': A', 'right': D'; 'D': 'Duity', 'D': { 'uft': B', 'left': 'C'} 'D': 'Duity', 'D': { 'uft': B', 'left': 'C'}