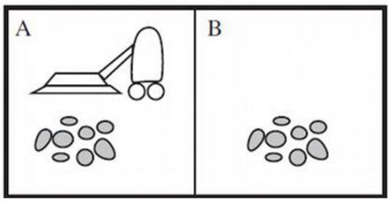
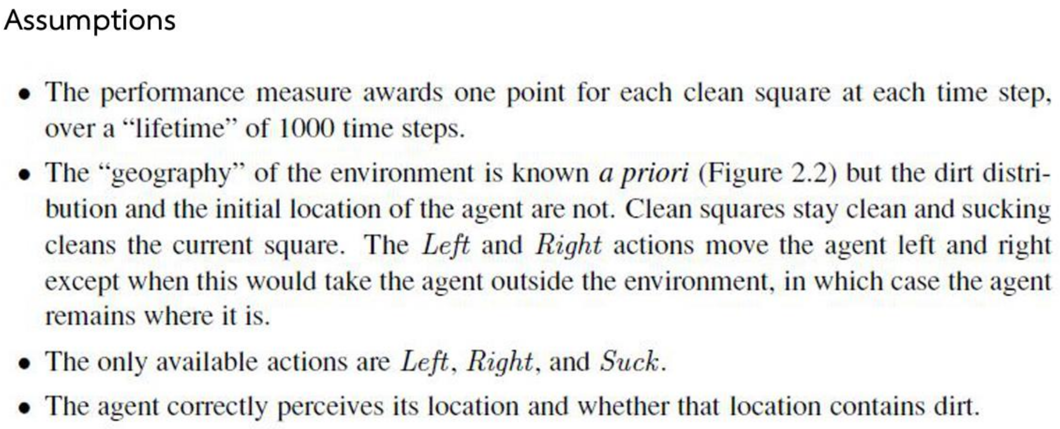
Q1: Implement a simulator of simple reflex agent for the vacuum cleaner environment as shown below and with the assumptions that follow.



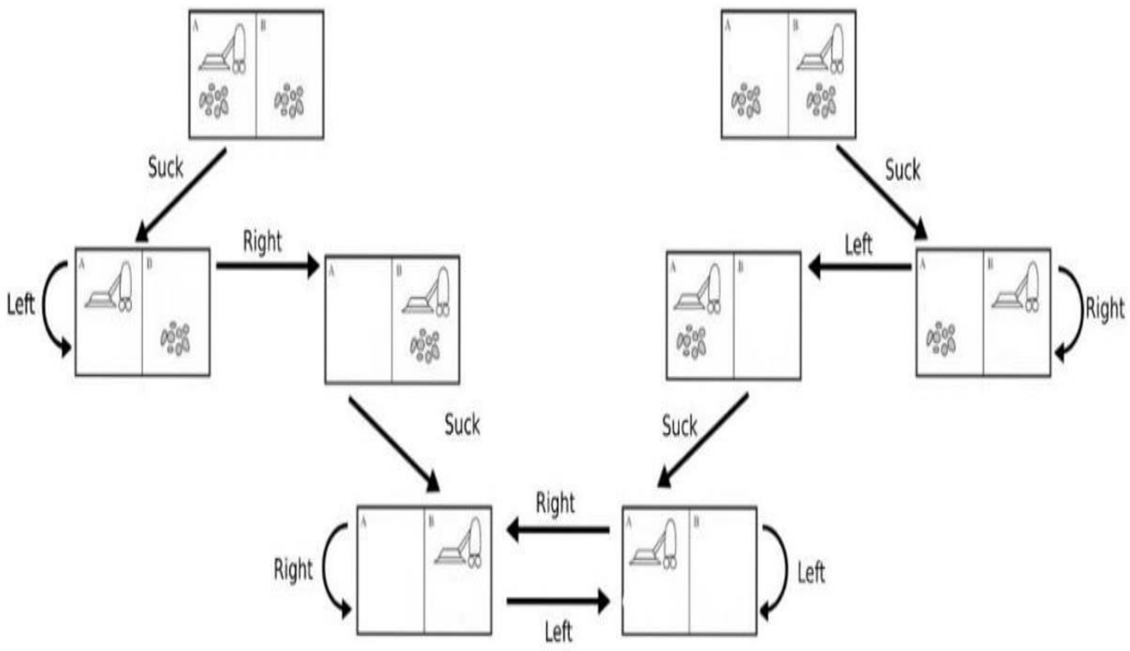


Assumptions

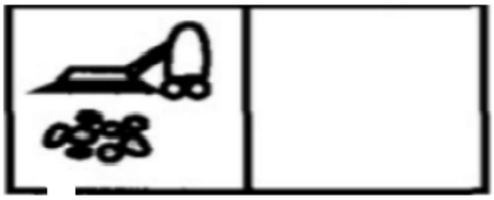
- The performance measure awards one point for each clean square at each time step, over a "lifetime" of 1000 time steps.
- The "geography" of the environment is known *a priori* (Figure 2.2) but the dirt distribution and the initial location of the agent are not. Clean squares stay clean and sucking cleans the current square. The *Left* and *Right* actions move the agent left and right except when this would take the agent outside the environment, in which case the agent remains where it is.
- The only available actions are *Left*, *Right*, and *Suck*.
- The agent correctly perceives its location and whether that location contains dirt.

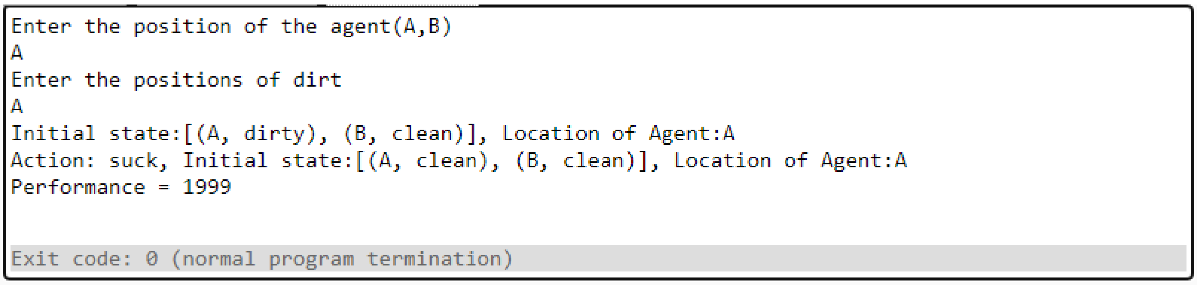
Ans:- Our Simulator has total of 1000 steps which includes actions like suck or move in left/right. Movement should be within the environment. At first, the simulator senses whether there is dirt in the location, if dirt is present, it sucks out the dirt. If the location is clean, it moves to left or right depending on which move does not take it out of the environment. 1 point is added to performance for each clean location present at each time step. Simulator prints the initial location, agent location and state of the environment at each time step. At the end it returns the performance of the agent.

State Diagram :-



Example 1:

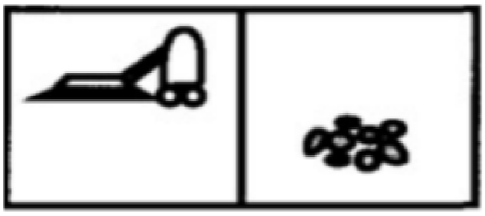


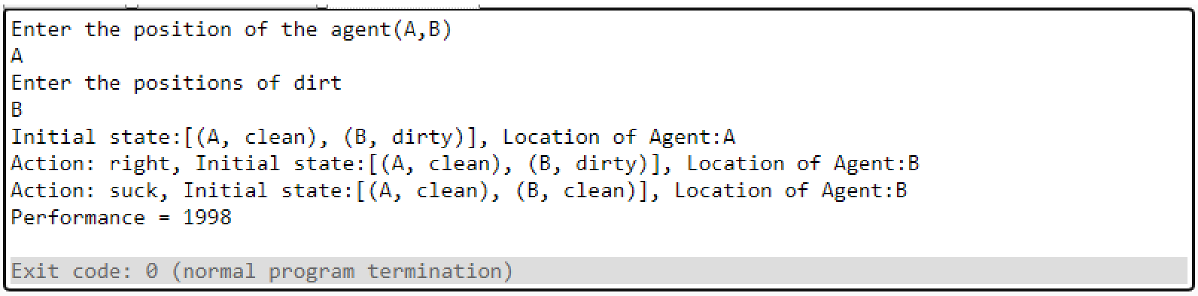


```
Enter the position of the agent(A,B)
A
Enter the positions of dirt
A
Initial state:[(A, dirty), (B, clean)], Location of Agent:A
Action: suck, Initial state:[(A, clean), (B, clean)], Location of Agent:A
Performance = 1999

Exit code: 0 (normal program termination)
```

Example 2:

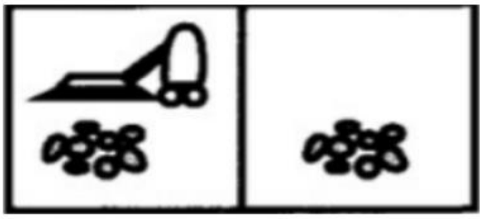


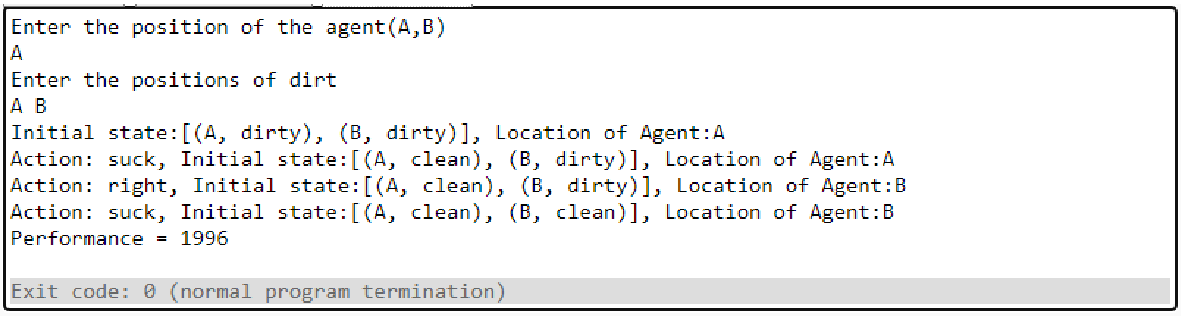


```
Enter the position of the agent(A,B)
A
Enter the positions of dirt
B
Initial state:[(A, clean), (B, dirty)], Location of Agent:A
Action: right, Initial state:[(A, clean), (B, dirty)], Location of Agent:B
Action: suck, Initial state:[(A, clean), (B, clean)], Location of Agent:B
Performance = 1998

Exit code: 0 (normal program termination)
```

Example 3:

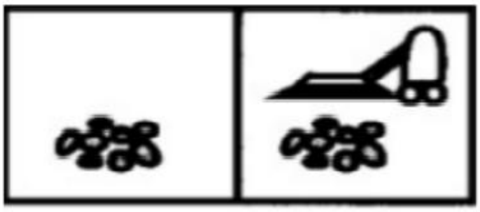


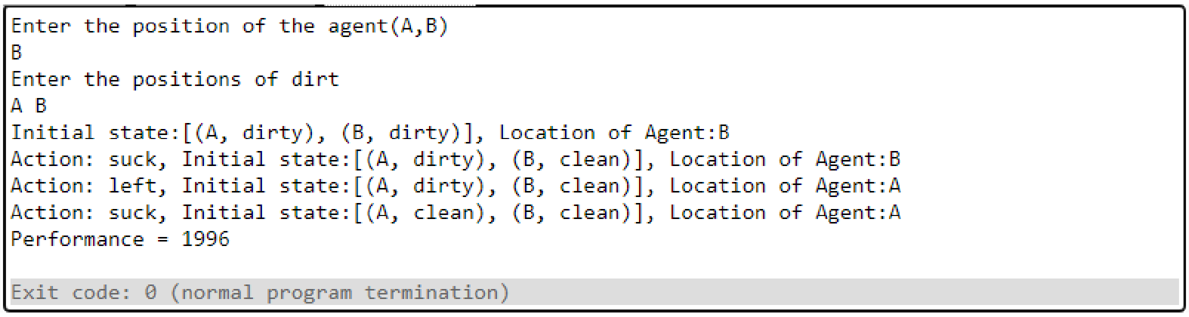


```
Enter the position of the agent(A,B)
A
Enter the positions of dirt
A B
Initial state:[(A, dirty), (B, dirty)], Location of Agent:A
Action: suck, Initial state:[(A, clean), (B, dirty)], Location of Agent:A
Action: right, Initial state:[(A, clean), (B, dirty)], Location of Agent:B
Action: suck, Initial state:[(A, clean), (B, clean)], Location of Agent:B
Performance = 1996

Exit code: 0 (normal program termination)
```

Example 4:

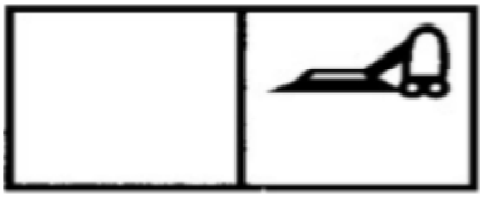


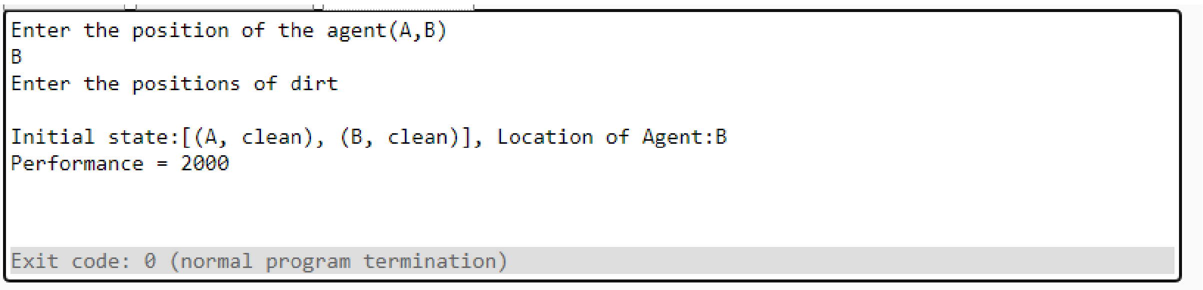


```
Enter the position of the agent(A,B)
B
Enter the positions of dirt
A B
Initial state:[(A, dirty), (B, dirty)], Location of Agent:B
Action: suck, Initial state:[(A, dirty), (B, clean)], Location of Agent:B
Action: left, Initial state:[(A, dirty), (B, clean)], Location of Agent:A
Action: suck, Initial state:[(A, clean), (B, clean)], Location of Agent:A
Performance = 1996

Exit code: 0 (normal program termination)
```

Example 5:





```
Enter the position of the agent(A,B)
B
Enter the positions of dirt

Initial state:[(A, clean), (B, clean)], Location of Agent:B
Performance = 2000



Exit code: 0 (normal program termination)
```