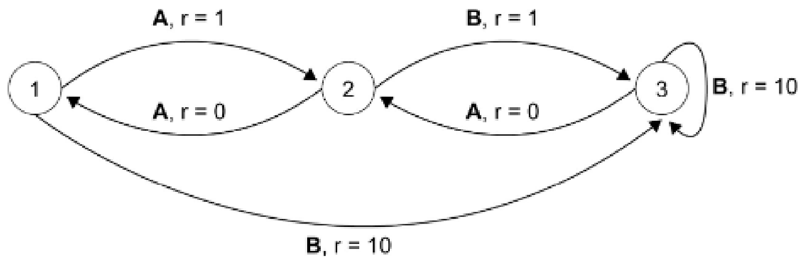
REINFORCEMENT LEARNING
LAB - 1

Deepak Yadav

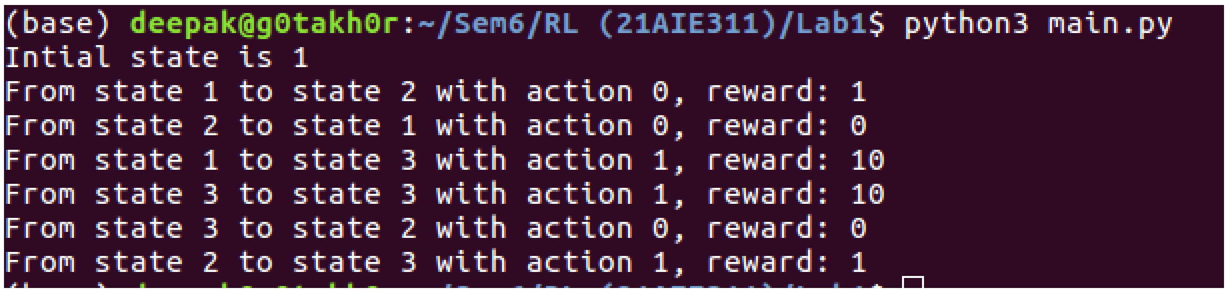
Part I

Figure given below shows a simple example composed of three state and two actions



ans)
#code

```
from typing import Tuple

class Environment:
    def __init__(self):
        self._initial_state=1
        self._allowed_action=[0,1]
        self._states=[1,2,3]
        self._current_state=self._initial_state

    def step(self,action:int)->Tuple[int,int]:
        if action not in self._allowed_action:
            raise ValueError("Action is not allowed")

        reward=0
        if action==0 and self._current_state==1:
            self._current_state=2
            reward=1
            return (self._current_state,reward)
        elif action==1 and self._current_state==1:
            self._current_state=3
            reward=10
            return (self._current_state,reward)
        elif action==0 and self._current_state==2:
            self._current_state=1
            reward=0
            return (self._current_state,reward)
        elif action==1 and self._current_state==2:
            self._current_state=3
            reward=1
            return (self._current_state,reward)
        elif action==0 and self._current_state==3:
            self._current_state=2
            reward=0
            return (self._current_state,reward)
        elif action==1 and self._current_state==3:
            self._current_state=3
            reward=10
            return (self._current_state,reward)

    def reset(self)->int:
        self._current_state=self._initial_state
        return self._current_state

env=Environment()
state=env.reset()

actions=[0,0,1,1,0,1]

print(f"Intial state is {state}")

for action in actions:
    next_state, reward=env.step(action)
    print(f"From state {state} to state {next_state} with action {action}, reward: {reward}")
    state=next_state
```
#Output



```
(base) deepak@g0takh0r:~/Sem6/RL (21AIE311)/Lab1$ python3 main.py
Intial state is 1
From state 1 to state 2 with action 0, reward: 1
From state 2 to state 1 with action 0, reward: 0
From state 1 to state 3 with action 1, reward: 10
From state 3 to state 3 with action 1, reward: 10
From state 3 to state 2 with action 0, reward: 0
From state 2 to state 3 with action 1, reward: 1
```

Part II

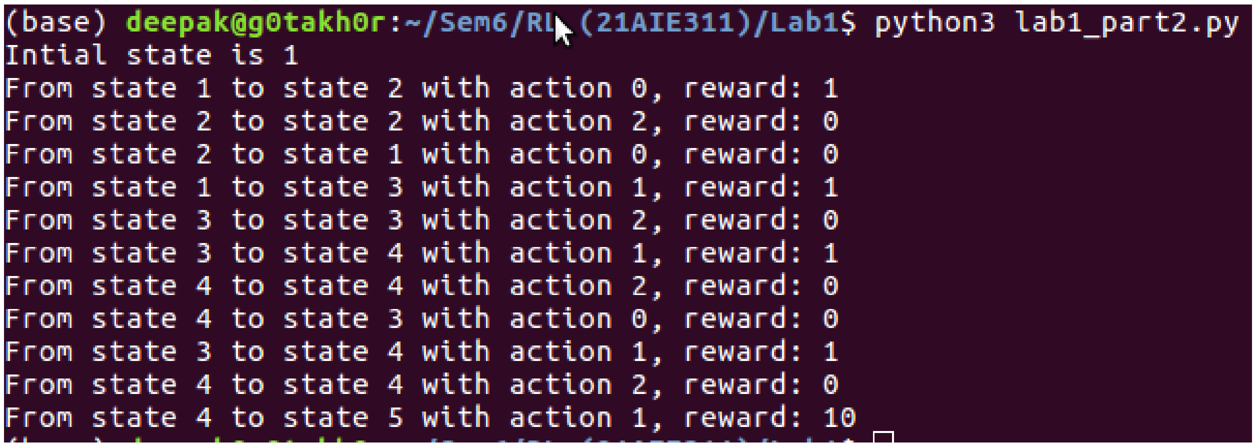
Creating a complex structure code with 5 states and 3 actions

ans)

#code

```
from typing import Tuple

class Environment:
    def __init__(self):
        self._initial_state=1
        self._allowed_action=[0,1,2]
        self._states=[1,2,3,4,5]
        self._current_state=self._initial_state

    def step(self,action:int)->Tuple[int,int]:
        if action not in self._allowed_action:
            raise ValueError("Action is not allowed")

        reward=0
        if action==0 and self._current_state==1:
            self._current_state=2
            reward=1
            return (self._current_state,reward)
        elif action==1 and self._current_state==1:
            self._current_state=3
            reward=1
            return (self._current_state,reward)
        elif action==2 and self._current_state==1:
            self._current_state=1
            reward=0
            return (self._current_state,reward)
        elif action==0 and self._current_state==2:
            self._current_state=1
            reward=0
            return (self._current_state,reward)
        elif action==1 and self._current_state==2:
            self._current_state=3
            reward=1
            return (self._current_state,reward)
        elif action==2 and self._current_state==2:
            self._current_state=2
            reward=0
            return (self._current_state,reward)
        elif action==0 and self._current_state==3:
            self._current_state=2
            reward=0
            return (self._current_state,reward)
        elif action==1 and self._current_state==3:
            self._current_state=4
            reward=1
            return (self._current_state,reward)
        elif action==2 and self._current_state==3:
            self._current_state=3
            reward=0
            return (self._current_state,reward)
        elif action==0 and self._current_state==4:
            self._current_state=3
            reward=0
            return (self._current_state,reward)
        elif action==1 and self._current_state==4:
            self._current_state=5
```

```
                reward=10
                return (self._current_state,reward)
            elif action==2 and self._current_state==4:
                self._current_state=4
                reward=0
                return (self._current_state,reward)
            elif action==0 and self._current_state==5:
                self._current_state=4
                reward=0
                return (self._current_state,reward)
            elif action==1 and self._current_state==5:
                self._current_state=5
                reward=10
                return (self._current_state,reward)
            elif action==2 and self._current_state==5:
                self._current_state=3
                reward=0
                return (self._current_state,reward)

    def reset(self)->int:
        self._current_state=self._initial_state
        return self._current_state

env=Environment()
state=env.reset()

actions=[0,2,0,1,2,1,2,0,1,2,1]

print(f"Intial state is {state}")

for action in actions:
    next_state, reward=env.step(action)
    print(f"From state {state} to state {next_state} with action {action}, reward: {reward}")
    state=next_state
#Output
```



```
(base) deepak@g0takh0r:~/Sem6/RL (21AIE311)/Lab1$ python3 lab1_part2.py
Intial state is 1
From state 1 to state 2 with action 0, reward: 1
From state 2 to state 2 with action 2, reward: 0
From state 2 to state 1 with action 0, reward: 0
From state 1 to state 3 with action 1, reward: 1
From state 3 to state 3 with action 2, reward: 0
From state 3 to state 4 with action 1, reward: 1
From state 4 to state 4 with action 2, reward: 0
From state 4 to state 3 with action 0, reward: 0
From state 3 to state 4 with action 1, reward: 1
From state 4 to state 4 with action 2, reward: 0
From state 4 to state 5 with action 1, reward: 10
```

Part III

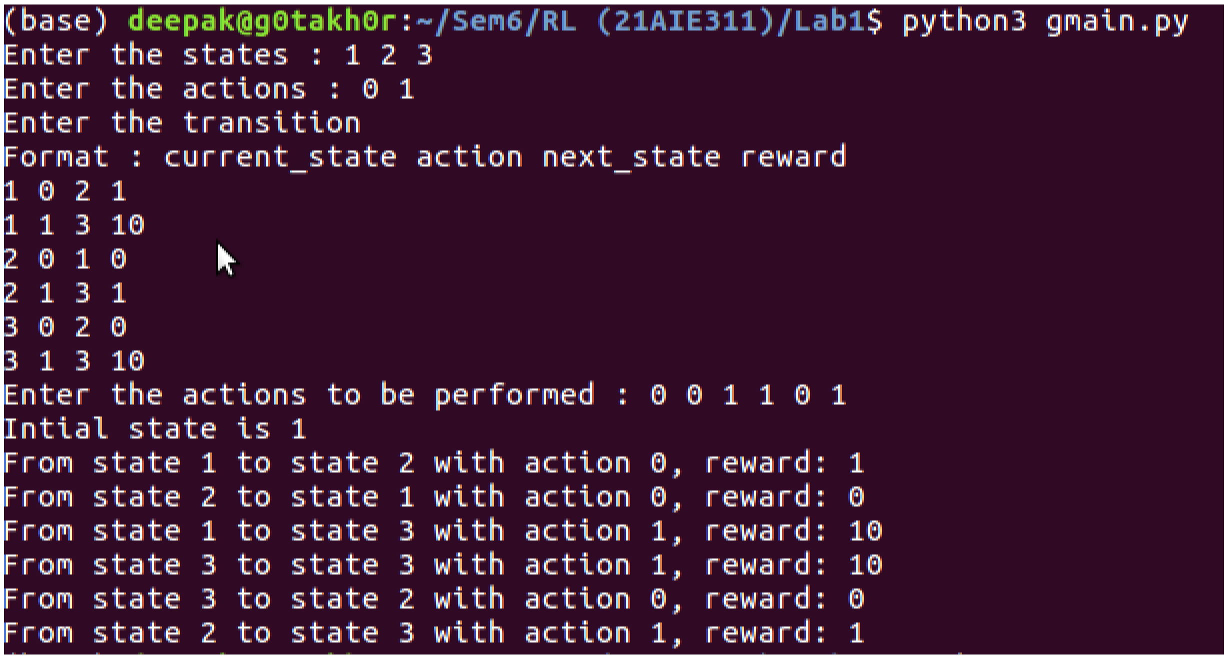
Generalizing the problem with n state and m actions

ans)

Performing the same model demonstration as the first one

```
#code
from typing import Tuple

class Environment:
    def __init__(self,_allowed_action,_states):
        self.trf = {}
        self._initial_state=1
        self._allowed_action=_allowed_action
        self._states=_states
        self._current_state=self._initial_state
        print("Enter the transition")
        print("Format : current_state action next_state reward")
        for i in range((len(_allowed_action)*len(_states))):
            new=input().split(' ')
            new=[int(x) for x in new]
            s, a, n, r = new
            self.trf[s,a]=(n, r)

    def step(self,action:int)->Tuple[int,int]:
        perform = self.trf[(self._current_state,action)] #matching the dictonary to perform
action
```

```
            if perform:
                n,r=perform
                self._current_state=n
                return (self._current_state,r)
            else:
                print("Error!!!")
                return 0

    def reset(self)->int:
            self._current_state=self._initial_state
            return self._current_state

st=input("Enter the states : ").split()
st=[int(x) for x in st]
ac=input("Enter the actions : ").split()
ac=[int(x) for x in ac]
env=Environment(st,ac)

state=env.reset()

act=input("Enter the actions to be performed : ").split()
act=[int(x) for x in act]
actions=act

print(f"Intial state is {state}")

for action in actions:
    next_state, reward=env.step(action)
    print(f"From state {state} to state {next_state} with action {action}, reward: {reward}")
    state=next_state

#Output
```



```
(base) deepak@g0takh0r:~/Sem6/RL (21AIE311)/Lab1$ python3 gmain.py
Enter the states : 1 2 3
Enter the actions : 0 1
Enter the transition
Format : current_state action next_state reward
1 0 2 1
1 1 3 10
2 0 1 0
2 1 3 1
3 0 2 0
3 1 3 10
Enter the actions to be performed : 0 0 1 1 0 1
Intial state is 1
From state 1 to state 2 with action 0, reward: 1
From state 2 to state 1 with action 0, reward: 0
From state 1 to state 3 with action 1, reward: 10
From state 3 to state 3 with action 1, reward: 10
From state 3 to state 2 with action 0, reward: 0
From state 2 to state 3 with action 1, reward: 1
```