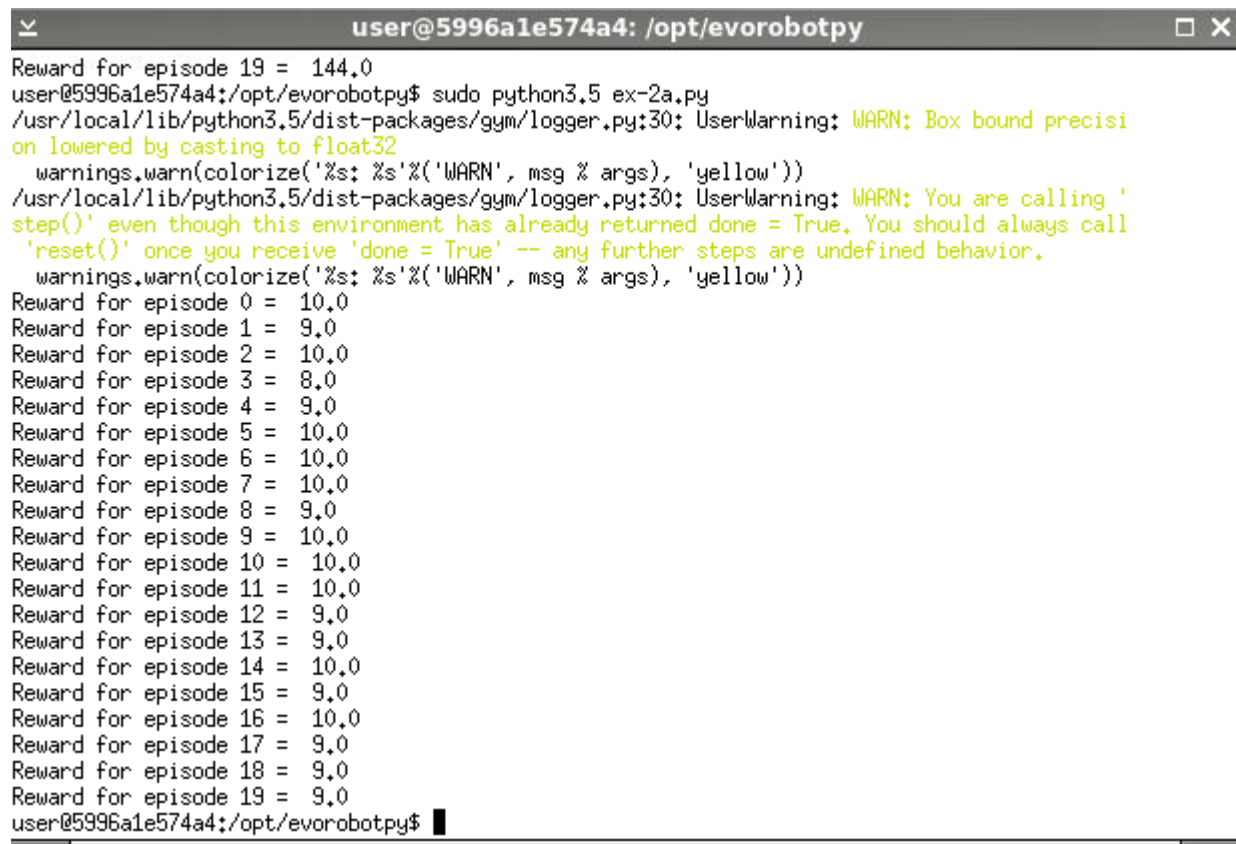


Exercise 2A

The task is to implement neural network and interact it with cart-pole system to stabilize it. The file `ex-2a.py` contains the solution of the task. In the first part, the weights and biases of the neural network are initialized randomly. It is important to note that in this case the cart-pole environment is not simulated with random guess but the neural network is used to return action object that is later used to render the environment. In each step, the activation of neurons is calculated and environment is simulated. The fitness function in cart-pole is used to evaluate the performance of the neural network. The fitness function is nothing but the cumulative reward for each episode. The following figure shows the result of code.



```
user@5996a1e574a4: /opt/evorobotpy
Reward for episode 19 = 144.0
user@5996a1e574a4:/opt/evorobotpy$ sudo python3.5 ex-2a.py
/usr/local/lib/python3.5/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%( 'WARN', msg % args), 'yellow'))
/usr/local/lib/python3.5/dist-packages/gym/logger.py:30: UserWarning: WARN: You are calling 'step()' even though this environment has already returned done = True. You should always call 'reset()' once you receive 'done = True' -- any further steps are undefined behavior.
  warnings.warn(colorize('%s: %s'%( 'WARN', msg % args), 'yellow'))
Reward for episode 0 = 10.0
Reward for episode 1 = 9.0
Reward for episode 2 = 10.0
Reward for episode 3 = 8.0
Reward for episode 4 = 9.0
Reward for episode 5 = 10.0
Reward for episode 6 = 10.0
Reward for episode 7 = 10.0
Reward for episode 8 = 9.0
Reward for episode 9 = 10.0
Reward for episode 10 = 10.0
Reward for episode 11 = 10.0
Reward for episode 12 = 9.0
Reward for episode 13 = 9.0
Reward for episode 14 = 10.0
Reward for episode 15 = 9.0
Reward for episode 16 = 10.0
Reward for episode 17 = 9.0
Reward for episode 18 = 9.0
Reward for episode 19 = 9.0
user@5996a1e574a4:/opt/evorobotpy$
```

Figure. 1

The weights of the neural networks are initialized randomly so the reward is not appreciable for each step. In the next exercise the network is fine tuned to obtain good results.