CMPUT 366 P1 Report
Sean Scheideman
Marcin Pietrasik

1.
The observed average return using an equiprobable random policy was about -0.33 which is close to the expected value. To implement the random policy we used python random.randint(0,1).

2.
To check if our Expected Sarsa was implemented correctly we ran it with epsilonu = 1 to start, in which case we got the approximately -0.33 for average return which matched the equiprobable random policy from part 1.

Next we used the settings specified in the assignment and got the following output:

alpha = 0.001, epsilonu = 0.01 and epsilonpi = 0.01

```
Count = 10000 Average return:  -0.1148
Count = 20000 Average return:  -0.09745
Count = 30000 Average return:  -0.092
Count = 40000 Average return:  -0.09045
Count = 50000 Average return:  -0.0868
Count = 60000 Average return:  -0.0851166666667
Count = 70000 Average return:  -0.0852142857143
Count = 80000 Average return:  -0.0845125
Count = 90000 Average return:  -0.0819888888889
Count = 100000 Average return:  -0.08024
Count = 110000 Average return:  -0.0789181818182
Count = 120000 Average return:  -0.0794666666667
Count = 130000 Average return:  -0.0806230769231
Count = 140000 Average return:  -0.0799785714286
Count = 150000 Average return:  -0.07914
Count = 160000 Average return:  -0.07785625
Count = 170000 Average return:  -0.0777176470588
Count = 180000 Average return:  -0.0779222222222
Count = 190000 Average return:  -0.0766052631579
Count = 200000 Average return:  -0.075435
Count = 210000 Average return:  -0.0744523809524
Count = 220000 Average return:  -0.0735363636364
Count = 230000 Average return:  -0.0735130434783
Count = 240000 Average return:  -0.073825
Count = 250000 Average return:  -0.07298
Count = 260000 Average return:  -0.0720307692308
Count = 270000 Average return:  -0.0719851851852
Count = 280000 Average return:  -0.0721285714286
Count = 290000 Average return:  -0.0717137931034
Count = 300000 Average return:  -0.0713066666667
Count = 310000 Average return:  -0.0706548387097
Count = 320000 Average return:  -0.06989375
Count = 330000 Average return:  -0.0693333333333
Count = 340000 Average return:  -0.0691411764706
```

```
Count = 350000 Average return:  -0.0685971428571
Count = 360000 Average return:  -0.0679277777778
Count = 370000 Average return:  -0.0680594594595
Count = 380000 Average return:  -0.0675894736842
Count = 390000 Average return:  -0.0670897435897
Count = 400000 Average return:  -0.0662575
Count = 410000 Average return:  -0.0655268292683
Count = 420000 Average return:  -0.0653595238095
Count = 430000 Average return:  -0.0653
Count = 440000 Average return:  -0.0651272727273
Count = 450000 Average return:  -0.0644777777778
Count = 460000 Average return:  -0.0636347826087
Count = 470000 Average return:  -0.0635
Count = 480000 Average return:  -0.0630770833333
Count = 490000 Average return:  -0.0631367346939
Count = 500000 Average return:  -0.06315
Count = 510000 Average return:  -0.062731372549
Count = 520000 Average return:  -0.0621403846154
Count = 530000 Average return:  -0.0619358490566
Count = 540000 Average return:  -0.0613277777778
Count = 550000 Average return:  -0.0612236363636
Count = 560000 Average return:  -0.0609446428571
Count = 570000 Average return:  -0.060498245614
Count = 580000 Average return:  -0.0601086206897
Count = 590000 Average return:  -0.0599762711864
Count = 600000 Average return:  -0.0596983333333
Count = 610000 Average return:  -0.0594770491803
Count = 620000 Average return:  -0.0590274193548
Count = 630000 Average return:  -0.0586317460317
Count = 640000 Average return:  -0.0583234375
Count = 650000 Average return:  -0.0579446153846
Count = 660000 Average return:  -0.0574893939394
Count = 670000 Average return:  -0.0571417910448
Count = 680000 Average return:  -0.0568647058824
Count = 690000 Average return:  -0.056552173913
Count = 700000 Average return:  -0.0562142857143
Count = 710000 Average return:  -0.0562394366197
Count = 720000 Average return:  -0.0561944444444
Count = 730000 Average return:  -0.0558917808219
Count = 740000 Average return:  -0.0559054054054
Count = 750000 Average return:  -0.055692
Count = 760000 Average return:  -0.0553881578947
Count = 770000 Average return:  -0.0553233766234
Count = 780000 Average return:  -0.0550038461538
Count = 790000 Average return:  -0.0546696202532
Count = 800000 Average return:  -0.0545175
Count = 810000 Average return:  -0.0543814814815
Count = 820000 Average return:  -0.0542292682927
Count = 830000 Average return:  -0.0541
Count = 840000 Average return:  -0.0539035714286
Count = 850000 Average return:  -0.0537647058824
Count = 860000 Average return:  -0.0536430232558
Count = 870000 Average return:  -0.053424137931
Count = 880000 Average return:  -0.0534545454545
```

```
Count = 890000 Average return:  -0.053408988764
Count = 900000 Average return:  -0.0533088888889
Count = 910000 Average return:  -0.0530989010989
Count = 920000 Average return:  -0.0529065217391
Count = 930000 Average return:  -0.0528516129032
Count = 940000 Average return:  -0.0528361702128
Count = 950000 Average return:  -0.0527631578947
Count = 960000 Average return:  -0.0526083333333
Count = 970000 Average return:  -0.0523886597938
Count = 980000 Average return:  -0.0522979591837
Count = 990000 Average return:  -0.0521676767677

 Usable Ace:
S H H S H S H S H S 20
S H H H S S S S S S 19
S H H H H H H H S S 18
S S S H H H S H H H 17
H H H H H H H S H H 16
S H H H H H H H H H 15
H H H H H H H H H H 14
H H H H H H H H H H 13
H H H H H H H H H H 12
1 2 3 4 5 6 7 8 9 10

 No Usable Ace:
S S S S S S S S S S 20
S S S S S S S S S S 19
S S S S S S S S S S 18
S S S S S S S S S S 17
H S S S S S H H H H 16
H H S H H H H H H H 15
H H S H H H H H H H 14
H H H H H H H H H H 13
H H H H H H H H H H 12
1 2 3 4 5 6 7 8 9 10
Average return:  -0.051992
```

Next we used the learned deterministic policy above without exploration and ran for 10 million episodes and got the following average return:

```
Average return deterministic:  -0.037333
```

3.
After experiment with various settings the combination we found were:

Alpha = 0.001, epsilonu = 0.19 and epsilonpi = 0.05

Below is the policy and average return found using the settings above for 10 million episodes.

```
 Usable Ace:
S S S S S S S S S 20
S S S S S S S S S 19
H S S S S S S H H 18
H H H H H S H H H H 17
H H H H H H H H H H 16
H H H H H H H H H H 15
H H H H H H H H H H 14
H H H H H H H H H H 13
H H H H H H H H H H 12
1 2 3 4 5 6 7 8 9 10

 No Usable Ace:
S S S S S S S S S 20
S S S S S S S S S 19
S S S S S S S S S 18
S S S S S S S S S 17
H S S S S S H H H H 16
H H H S S H H H H H 15
H H H S H H H H H H 14
H H H H H H H H H H 13
H H H H H H H H H H 12
1 2 3 4 5 6 7 8 9 10
```

Final performance level (average return): -0.0279839

Below are some of the test cases we ran to find the best values:

alpha	**eu**	**epi**	**Average**	
0.001	1	0.01	-0.027860	
0.001	0.0001	0.0001	-0.033427	
0.001	1	1	-0.043871	
0.001	0.19	0.19	-0.028777	
1	0.19	0.05	-0.154074	
0.1	0.19	0.05	-0.034892	
0.01	0.19	0.05	-0.029551	
0.001	0.19	0.05	-0.027012	(max)
0.0001	0.19	0.05	-0.030434	

Based off these values we found that a behaviour epsilon of about 0.19 and a policy epsilon of 0.05 produced the best policy consistently. With these parameters we are being more exploratory using the behaviour policy; meanwhile being greedier when updating the value function using the target policy and essentially using Q-Learning.