Name | Student Number
Oluwatomilayo Adegbite | 500569283
Nikolas Maier | 500461990

Policy and value iteration seem to have their individual pro's and con's.
Policy iteration has a high time complexity, a minimum of $O(n^2)$ due to having to go through all states at least once during policy evaluation and then a second time through all states for policy improvement. This ensures that policy iteration will take longer than value iteration and should be viewed as a negative.
A positive however is the low number of iterations. Due to so much data being checked again and again the number of iterations required by policy iteration is very low. In our experimenting it took 2 iterations for the policy to be classified stable.
The time and final policy of one test can be seen below.

```
C:\Users\tib-n>python.lnk "C:\Users\tib-n\Documents\GitHub\Enter-The-Grid\Niks WIP\UserInterface.py"
Policy Iteration:

Iteration: 1 has taken 0.0059833526611328125
Iteration: 2 has taken 0.001964092254638672
```

0	<	<	<
^	^	<	v
^	^	v	v
>	>	>	0

Value iteration has its own positives and negatives, which can be largely surmised based on the opposites of Policy Iteration. Iterations taken by this algorithm are far faster as it doesn't need to perform both Policy Evaluation and Policy Improvement. However the tradeoff is it requires more iterations to hammer out all smaller changes in numbers that can occur due to this difference and even though Value iteration does need more iterations through the states it winds up being much faster than policy iteration is.

```
Value Iteration:

Iteration: 1 has taken 0.0

Iteration: 2 has taken 0.000997304916381836

Iteration: 3 has taken 0.0

Iteration: 4 has taken 0.0

Iteration: 5 has taken 0.0009958744049072266
```

0	<	<	v
^	<	v	v
^	>	v	v
>	>	>	0