Question 1

The optimal policy says to move left because moving up has a 20% chance of failing to move up – that is, a 10% chance of moving left and a 10% chance of moving right. Notice that square (4, 1) has a very low utility and there is a 10% chance of falling into that square. To avoid that, we instead move left.

Here are the calculations for moving left:

$$\text{Utility} = \ 0.8 * 0.655 + 0.1 * 0.660 + 0.1 * 0.611 = 0.6511$$

Here are the calculations for moving up:

$$\text{Utility} = 0.8 * 0.660 + 0.1 * 0.655 + 0.1 * 0.388 = 0.6323$$

From these calculations, we see that the chance of getting 0.388 penalizes the action of moving up enough that it is more optimal to move left instead with no chance of getting 0.388.

Question 2

Here were the results of the Monte Carlo simulation:
10 run mean: 0.308
10 run stddev: 0.678126831795
100 run mean: 0.3128
100 run stddev: 0.657105897097
1000 run mean: 0.43424
1000 run stddev: 0.537426853069

The means were are close to 0.388, the utility of the (4, 1) square, which makes sense. The means are all pretty close together as well.

Looking at the histograms (especially at the 1000-run one), we can almost see a normal distribution about utility slightly more negative than -1 and another normal distribution about utility 0.66. It makes sense that there are two distributions because one would correspond to ending up in the +1 state and one would correspond to ending in the -1 state.

The path to the +1 state goes (4,1) -> (3,1) -> (2,1) -> (1,1) -> (2,1) -> (3,1) -> (3,2) -> (3,3) ->(3,4). Most paths that end at +1 would go through some path like this. This passes through 9 squares and the first 8 squares all cost -0.04 due to the cost of living. So, most paths that end at +1 would end up with a final reward of 1 – 0.32 which is about 0.66. There are a few lucky simulations where upon trying to move from (4,1) -> (3,1), the person gets pulled up to (3, 2) and luckily reaches (3, 3) and finally (3, 4). This explains the small distribution of rewards greater than 0.66. There are a few redundant simulations where upon trying to continue down the path, the person gets pulled in the perpendicular direction into a wall. This just increases the total cost of living and accounts for the small distribution of positive values less than 0.66.

There is a much smaller normal distribution around -1.1. The bell shape is shorter because by following the optimal policy, you are more likely to reach +1 than -1. There are no values in the range (-1.04, 0] because the cost of living is negative and ending up in the -1 square only makes the reward more negative. The peak at (-1.04, 3) corresponds to the times when the unlucky individual was pulled into the (4,2) square immediately when trying to move from (4,1) to (3,1). There is a small tail corresponding to the unlucky person falling into the -1 square by going through the paths (4,1), (3,1), (3,2), (4,2) and perhaps getting pulled from the (3,3) square down to (3,2) and getting pulled again from (3,2) to (4,2). These are all very unlikely events though, so the tail of very negative rewards is very small.

Problem 3

The discount rate represents how much you'd rather have the given reward now than in the future. A small discount rate (e.g. 0.01) means you really want rewards now. So, it makes sense for you to get as much positive reward as early as possible, and potentially ignore even higher rewards that take time to get to.

By increasing the discount rate in the problem, we make the optimal policy guide the person to the +3 square sooner and stay there forever. Notice that for a discount rate of about 0.7, the payoff from starting in the +3 square and staying there forever is the sum of the infinite sequence $3 + 0.7*3 + 0.7^2*3 + \ldots$, which converges to 10. So, at discount rates lower than 0.7, given the choice of moving to 3 versus 10, from square(2,2), the person actually choose to move to 10 rather than stay in 3 forever.

When the discount rate increases high enough, it is almost always optimal to take the +3 with a discount forever rather than the one-time payoff of +10. (i.e. a small inflow of positive payments over a one-time large, positive payment).

Problem 1 Output

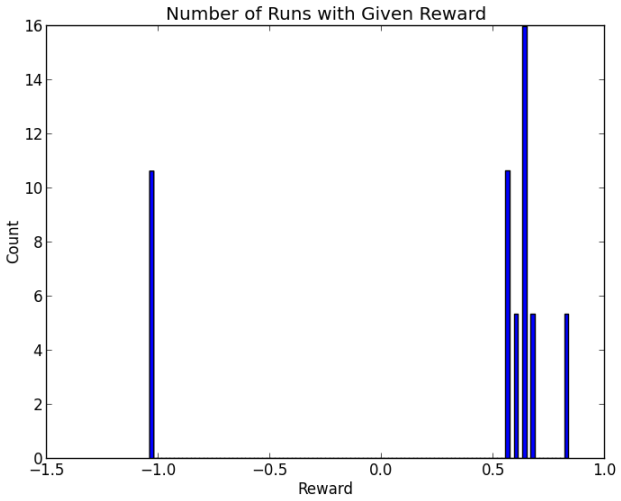
0.0000
> > > 1
^ X < -1
^ < < V
-0.0221
> > > 1
^ X < -1
^ < < <*
-0.0274
> > > 1
^ X ^* -1
^ < < <
-0.0448
> > > 1
^ X ^ -1
^ < ^* <
-0.0850
> > > 1
^ X ^ -1
^ >* ^ <
-0.4526
> > > 1
^ X ^ -1
^ > ^ ^*
-0.7312
> > > 1
^ X ^ -1
>* > ^ ^
-1.5643
> > > 1
^ X ^ -1
> > >* ^
-1.6497
> > > 1
^ X >* -1
> > > ^

Problem 2 Output

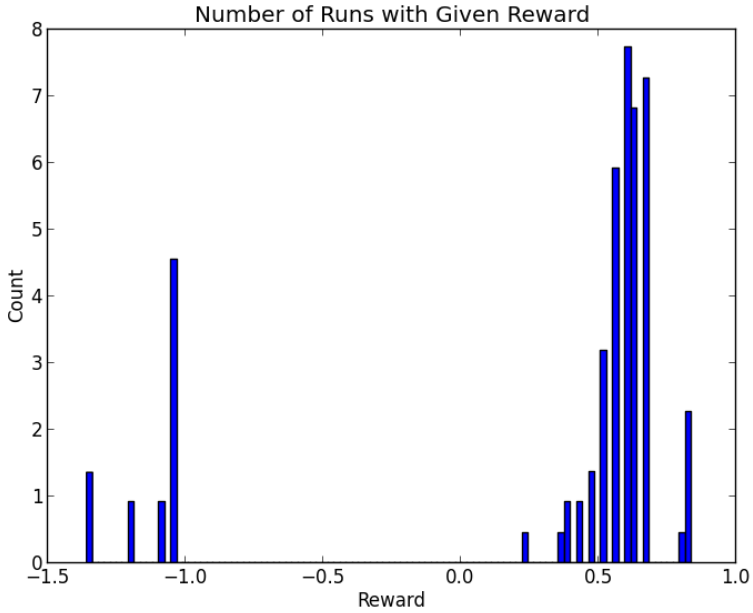
Utility Map:

```
0.812 0.868 0.918 1.000
0.762 0.000 0.660 -1.000
0.705 0.655 0.611 0.388
```

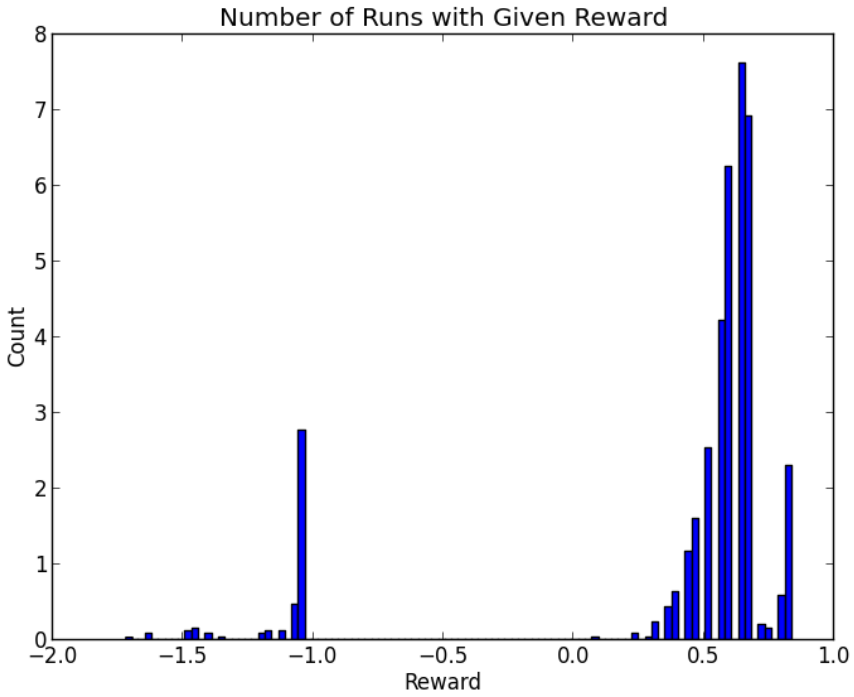
10-run histogram:



100-run histogram:



1000-run histogram:



Problem 3 Output

```
0.99000
^ < 10
^ < V
^ < <
87.143 82.264 10.000
82.264 78.085 67.413
77.635 74.083 69.880
0.97102
^ < 10
^ < <*
^ < <
21.026 16.297 10.000
16.297 12.709 9.427
12.364 9.635 7.220
0.87706
^ < 10
^ < <
^ ^* <
18.507 13.803 10.000
13.803 10.317 7.401
9.988 7.399 5.171
0.85976
^ < 10
^ ^* ^*
^ ^ ^*
18.518 13.814 10.000
13.814 10.327 7.409
9.999 7.409 5.179
0.85985
^ < 10
^ ^ <*
^ ^ ^
10.000 5.489 10.000
5.488 3.023 5.489
2.484 1.137 2.484
0.73308
^ >* 10
^ ^ ^*
^ ^ ^
3.000 -1.000 10.000
-1.000 -1.000 -1.000
-1.000 -1.000 -1.000
0.00000
```