CS1115 Lab 6 (October 3, 2013)

Completing the lab is very important, but your work is not graded and it is not submitted. If you finish before the hour is over, then you can leave early or you can work on the "Finished Early" problems. If you do not finish the problems before the end of class, then be sure to ask enough questions so that you can complete the exercises in the next day or two on your own.

1 Random Walk

Download the functions `RandWalk` and `ShowRandWalk` from the syllabus page. Run the latter and browse through the former to understand how the random walk works.

(a) For a given N, the token exit is "happy" if the token's final xy coordinates are either $(0, N)$, $(N, 0)$, $(-N, 0)$ or $(0, -N)$. Write a single script that estimates the probability of a happy exit for $N = 10$, $N = 15$, $N = 20$ and $N = 25$.

(b) Write a function `m = CrossThe45(N)` that assigns to `m` the number of times that the token actually crosses the 45-degree line on its way to the border. Your implementation should make use of `RandWalk`. Hint. If $(x(k), y(k))$ is on the 45 degree line, then there may be a crossing as the token moves from $(x(k-1), y(k-1))$ to $(x(k+1), y(k+1))$. What value is displayed when the following script is run?

```
N = 20;
rand('seed',0);
m = CrossThe45(N)
```

(c) The probability that the token hops north is twice the probability that it hops east, half the probability that it hops south, and the same as the probability that it hops west. Modify `RandWalk` accordingly and write a script that simultaes one-million random walks with $N = 20$ and reports the number of north edge exits, east edge exits, south edge exits, and west edge exits.

2 Finished Early?

The following fragment assigns to `deck` a length 52 column vector that is a random permutation of the integers 1 through 52:

```
[x,deck] = sort(rand(52,1))
```

(We learn about MATLAB's `sort` function in §8.2. For now, it is a "black box" method for generating random rearrangements of the first fifty-two integers.) Let us identify each integer with a playing card:

1	♠ Ace	14	♡ Ace	27	♣ Ace	40	♢Ace
2	♠ Two	15	♡ Two	28	♣ Two	41	♢ Two
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
13	♠ King	26	♡ King	39	♣ King	52	♢ King

Using boolean variables and vector comparisons we can readily answer "dealt hand" questions. For example, suppose your hand consists of the cards represented by `deck(1:13)`. Here is a statement that assigns to `nHearts` the number of hearts in your hand:

```
nHearts = sum( 14 <= deck(1:13) & deck(1:13) <= 26 )
```

Here is a statement that computes the number of Jacks in a 5-card hand:

```
nJacks = sum(mod(deck(1:5),13)==11);
```

(a) In poker, a 5-card hand with all the cards in the same suit is called a flush. Write a fragment that assigns to `IsFlush` the value of 1 if `deck(1:5)` is a flush and the value of 0 if it is not a flush. (b) Write a script that estimates the probablity that a 5-card hand is a flush. (c) Write a script that estimates the probability that in a 5-card hand no two cards have the same rank. (d) Write a script that estimates the probability that in a 5-card hand, at least one suit is missing. (e) Write a script that estimates the probability that in a 5-card hand, exactly three of the cards have the same rank.

Please delete your files from the computer before you leave the lab!