Assignment #5

Math 2604: Mathematical Programming in Python

Instructions: Choose 3 of the following problems to work on. Submit your responses as Python text files, with the extension `.py`. Each file should include your name and the problem number.

- *Problem 5.0:* The middle square method of generating a random sequence of integers of $2d$ digits was described as follows:

  1. Initialize s, an integer with $2d$ digits;
  2. Replace s by $s * s$, which has no more than $4d$ digits;
  3. Drop the last d digits of s by integer division by 10^d;
  4. Drop the first d digits of s by computing s modulo 10^{2d}.
  5. s is now the remaining middle $2d$ digits;
  6. To get another value, go to step 2.

  Implement this algorithm, with $d = 2$, starting with $s = 3647$ and reproduce the following sequence of s values:

i	s	s^2
0	3647	13300609
1	3006	9036036
2	360	129600
3	1296	1679616
4	6796	46185616
5	1856	3444736
6	4447	19775809
7	7758	60186564
8	1865	3478225
9	4782	22867524
10	8675	75255625

- *Problem 5.1:* In assignment #3, problem 5, you were asked to approximate π by sampling n random points (x, y) in the unit square, finding that k of these points were in the unit circle, and writing $\pi \approx \frac{4k}{n}$. Presumably, you used a `for()` loop to do this. Try to repeat this calculation using vectorized statements. You should need just two lines of calculation. In one line, compute all the (x, y) data. In the second line, estimate π using a statement that determines k. (Problem P2.5.10 from Hill.)

- *Problem 5.2:* Suppose you are given d, the index of a day of the (non-leap) year, with $0 \leq d < 365$. Suppose that `month` is a list of the names of the months:

```
months = [ 'January', 'February', ..., 'December' ]
```

  and that `month_length` is a `numpy` array of the number of days in each month:

```
month_length = np.array ( [ 31, 28, ..., 31 ] )
```

  Given a value for d, write a function that prints out the corresponding month and day. For instance, given $d = 68$, your function should print

```
day 68 is March 10.
```

  You might simply use a `for()` loop to do this. Or it might help to notice that the `numpy` function `y=cumsum(x)` returns a vector of the cumulative sums of the entries in `x`.

- *Problem 5.3:* A deck of cards has 4 suits: 'Hearts', 'Clubs', 'Diamonds', 'Spades', each suit containing 13 ranks, which we can label '1' through '13'. Assume the 1 of Hearts is the first card (number 0), and the 13 of Spades is the last (number 51). Write a function that is given a value c with $0 \leq c < 52$, which prints out the suit and rank. For instance, given $c = 29$, your function should print

```
card 29 is 4 of Diamonds.
```

- *Problem 5.4:* As discussed in the notes on random numbers, the Newton-Pepys dice problem compares three processes:

  1. Six fair dice are tossed independently and at least one "6" appears.
  2. Twelve fair dice are tossed independently and at least two "6"s appear.
  3. Eighteen fair dice are tossed independently and at least three "6"s appear.

  Estimate the probability of success for each of the three cases. Newton thought case 1 was most likely to happen, while Pepys voted for case 3.

- *Problem 5.5:* As discussed in the notes on random numbers, when 5 cards are randomly selected from a deck of 52, a straight occurs if the five cards can be arranged to show consecutive rank, such as 8, 9, 10, 11, 12. The suits of the cards are ignored. Use the suggestions in the notes to estimate the chances of being dealt a straight, and compare your estimate to the theoretical value of $10,240/2,598,960 \approx 0.0039$.

- *Problem 5.6:* A *flush* in poker is a hand in which all 5 cards have the same suit. Using our numbering scheme, for instance, the cards with index 14, 18, 19, 22, 25 would represent a flush, consisting of 5 cards of the club suit. Estimate the probability of being dealt a flush in poker. The theoretical probability is $5108/2,598,960 \approx 0.001965$

- *Problem 5.7:* 25 people attend a party. Estimate the probabilty that at least two of these people have the same birthday. Hint: use `np.random.choice()` with `replace = True` to create a sample `b` of 25 birthdays with possible repetitions. Let `u = np.unique ( b )` return the unique items in this array. If the length of `u` is less than 25, at least one birthday is shared. Count how many times you find at least one shared birthday in 1000 cases to estimate the probability.

- *Problem 5.8:* A country decides to try to increase the female population by controlling births. If a mother gives birth to a daughter, she is permitted to have another child next year. If she gives birth to a boy, she is not allowed to have any more children. Thus, in this country, there will be never be a family with more than one boy, but there will be families with three, six, even ten daughters. Consider 1,000 families, each of whom has continued to have a child each year until a boy is born. Report:

  – The maximum number of children any family had;
  – The average number of children in a family;
  – The ratio of boys and girls born.

- *Problem 5.9:* Start a sequence by generating a random number r_1 between 0 and 1. Now, for $i = 2, 3, ...$, get a random number r_i, continuing as long as each value is larger than the previous one. Suppose n is the index of the random number r_n that first fails the test, that is, $r_n <= r_{n-1}$. What is the average value of this index n? Generate 10000 sequences this way to estimate this average. The theoretical average is $e \approx 2.7183...$