PHYS 160 - Exam #1 - Central Limit theorem

HINTS: This exam is doubled-sided. Don't forget your name or plot titles. If you are having trouble with a command (say display, or logplot) did you remember to import the proper library?

Normal Distribution (50pts)

- Create a function of three variables: $G(\mu, \sigma, x)$

$$G(\mu, \sigma, x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \quad (1)$$

- This is known as the normal distribution or a normalized Gaussian. When $\mu = 0, \sigma = 1$ it is the standard normal distribution (bell curve). Check to make sure that your function evaluates correctly. Print the decimal value of $G(0, 1, 1)$ (you should get $\approx .242$).
- Plot $G(0, 1, x)$ over $x = [-3..3]$ and check that this looks like the familar bell curve
- Evaluate as a decimal $G(0, 1, x)$ at the integer values $x = [-3, -2, ..., 2, 3]$.
- On a single plot, show $G(0, 1, x), G(0, 1/2, x), G(0, 1/3, x)$ over $x = [-3..3]$, ie. a Gaussian with $\sigma = [1, 1/2, 1/3]$.
- Use a log-plot to display $G(0, 1, x)$ over $x = [-3..3]$. Describe the resulting plot and the reason for its shape as a comment.
- Animate $G(0, \sigma, x)$, $x = [-10..10]$ over the $\sigma = [1..5]$.

Random Numbers (25 pts)

- Using the `with` command, import the following libraries `Statistics` and `RandomTools[MersenneTwister]`
- Define the following functions, paying careful attention to the syntax:

```
R := (total) -> [seq(GenerateInteger(range = 1 .. 6),n=1..total)];
S := (total) ->  add(GenerateInteger(range = 1 .. 6),n=1..total);
```

  The function $R(n)$ creates a list of n simulated throws of a regular six-sided die, while $S(n)$ calculates the sum those n throws. Try out a few examples to get the hang of it.

- Assign the variable $N = 5000$.
- Compute the mean, μ of N throws of two-dice. Do not use the `sum` command, but rather the `add` command to compute the summation (add has the same syntax as sum).

$$\mu 2 = \frac{1}{N} \sum_{i=1}^{N} S(2) \quad (2)$$

  As a check, your answer should be close to 7.

- Now compute the mean of N throws of 100 dice, save it as $\mu 100$:

$$\mu 100 = \frac{1}{N} \sum_{i=1}^{N} S(100) \quad (3)$$

Central Limit Theorem (25 pts)

- Using the `seq` command, make a list of length N and assign it to a variable `SLIST1`. This list should contain N instances of $S(1)$.
- Use the command `Histogram(SLIST1);` to view a histogram of this list. Describe the shape you see.
- Using the `seq` command, make a list of length N and assign it to a variable `SLIST100`. This list should contain N instances of $S(100)$.
- Use the command `Histogram(SLIST100);` to view a histogram of this list. Does the shape of it look familiar?
- On the same plot (remember the `display` command), plot both your histogram and $G(\mu 100, \sigma 100, x)$ over $x = [300..400]$. In this example use $\sigma 100 = 17$. If everything was done correctly they should lie nearly on top of each other.

The Central Limit theorem is deep, powerful, and often surprising result in physics and mathematics. In this simple example you've simulated a random process, namely throwing a die 100 times and adding up the result; when you plot the distribution of 5000 of these throws, they look nearly indistinguishable from a Gaussian. In fact, the sum of almost any random process added up looks like a Gaussian ... even if the original distribution looks unlike a Gaussian. Compare the histogram of a single die throw, SLIST1 versus that of 100 throws SLIST100 to see the dramatic difference.