Question 1:

You would like to survey students at Brown to determine the mean time $t$ (in hours) that they sleep each night. Assume you will independently sample $n$ students from the population, and let $X_i$ be the number of hours reported by student $i$. The mean of their answers is then

$$M_n = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

(a) As an initial guess, you assume the standard deviation of $X_i$ is 2.0 hours ($\text{Var}(X_i) = 4.0$). How large should $n$ be so that Chebyshev’s inequality guarantees that the estimate $M_n$ is within 30 minutes of the true $t$, with probability at least 0.99?

(b) You learn that all students have sleep times between 4 and 10 hours, and thus decide to set $\text{Var}(X_i) = (10 - 4)^2 / 12 = 3.0$, corresponding to a uniform distribution over that range. How should the value of $n$ obtained in part (a) be revised?

After learning about the central limit theorem, you assume that the number of surveyed students $n$ is sufficiently large that the distribution of $M_n$ is normal. The Matlab \texttt{norminv} command may be helpful when answering this question.

(c) Suppose you assume that the variance of $X_i$ is 4.0 hours, as in part (a). Assuming the distribution of $M_n$ is normal, how large should $n$ be so that the estimate $M_n$ is within 30 minutes of the true $t$, with probability at least 0.99?

(d) Suppose you assume that the variance of $X_i$ is 3.0 hours, as in part (b). Assuming the distribution of $M_n$ is normal, how large should $n$ be so that the estimate $M_n$ is within 30 minutes of the true $t$, with probability at least 0.99?

(e) Briefly discuss how and why the answers obtained using the central limit theorem differ from those obtained using Chebyshev’s inequality.
Question 2:

a) Your friend’s new puppy gets loose on Thayer street, and begins wandering aimlessly. Every minute, he travels south 5 meters with probability 1/2, or north 5 meters with probability 1/2. The directions of travel at successive minutes are independent. Using the central limit theorem, what is the approximate probability distribution of the puppy’s location after 1 hour? Where is he most likely to be?

b) Suppose that immediately after getting loose, the puppy smells food trucks to the south. Every minute, he travels south 5 meters with probability 2/3, or north 5 meters with probability 1/3. The directions of travel at successive minutes are independent. Using the central limit theorem, what is the approximate probability distribution of the puppy’s location after 1 hour? Where is he most likely to be?

c) In order to find the lost puppy, you and your friend decide to identify an interval of locations that contain the puppy with 95% probability. What is the length of this interval (in meters) for the motion model in part (a)? What is its length for the motion model in part (b)? Explain any differences.

Question 3:

Your friend has written a computer program that supposedly produces a sequence of independent random numbers that are uniformly distributed between 0 and 1. To test this program, you ask it to generate \( n = 1000 \) random samples, and compute their average.

a) Suppose that you find that the average of the \( n = 1000 \) samples is 0.55. Using the central limit theorem, if the generated numbers truly had a uniform distribution, what is the approximate probability of their average being greater than or equal to 0.55? Do you believe that your friend’s random number generator is correct?

b) Suppose that you find that the average of the \( n = 1000 \) samples is 0.50, exactly equal to the true mean of the target uniform distribution. Does this give you confidence that your friend’s random number generator is correct? Why or why not?
Algorithm 1 Bubble Sort

1: **input**: A list $L$ of $n$ real numbers
2: **for** Each element $x \in L$ from $L(2)$ to $L(n)$ **do**
3:     **while** $x$ is less than the element preceding $x$ in $L$ **do**
4:         swap that element with $x$
5:     **end while**
6: **end for**
7: **return** The sorted list $L$

**Question 4:**

Given a list of $n$ distinct real numbers, a sorting algorithm seeks to reorder the list from smallest to largest. As comparisons between elements are relatively slow, sorting algorithms are commonly judged by the number of comparisons they make. Algorithm 1 is called bubble sort because each element in turn “bubbles” up through the sorted part of the list to its proper place. We also provide a Matlab implementation of the bubble sort algorithm.

\( a) \) Compute the maximum possible number of comparisons between list elements made by the deterministic bubble sort in Algorithm 1, in terms of $n$.

\( b) \) To avoid the worst case, randomized bubble sort first randomly permutes the list, and then runs bubble sort. Let $X$ be an integer random variable equal to the number of comparisons made by randomized bubble sort. Compute $E[X]$ in terms of $n$.

\( c) \) Monte Carlo sampling can be used to estimate the cumulative distribution function of $X$. Use Matlab to execute the provided randomizedBubbleSort 1000 times for lists of length $n = 10$, and plot the empirical CDF $F_X(x)$. Compute and report the average number of comparisons across your 1000 Monte Carlo trials.

\( d) \) Repeat part (c) for lists of length $n = 100$, and discuss how the distribution of comparisons changes as the list length grows.