Problem 1. (Enhanced; most major points in the chapter covered in this problem)
What is the chance that when you roll 8 dice, the sum will be from 20 to 25 inclusive?

a. Simulate the histogram shown of the totals of a roll of eight dice.
   (bind box (seq 1 6))
   (bind sums nil)
   (dotimes (i 10000)
     (push (sum (draw 8 box)) sums))
   (hist sums integers)
   (shadecolors 19.5 25.5 red)
   Note that as well as shading the bins, the last command will compute the cumulative percentage
   of the values in those bins: in my run, 26.41 percent of the values fell in that range.

Results will be similar to the histogram below:

b. Calculate the theoretical proportion of values that should fall into the box:
   Mean of box = 3.5  SD of box = 1.7078
   EV of sum of 8 dice = 8 * 3.5 = 28  SE of sum of 8 dice = (sqrt 8) * 1.7078 = 4.8305
   Critical values (z-scores of boundaries):
   \[ Z_1 = \frac{19.5 - 28}{4.8305} = -1.7597 \]  so (bind Z1 -1.7597)
   \[ Z_2 = \frac{25.5 - 28}{4.8305} = -0.5175 \]  so (bind Z2 -0.5175)
   Area under the normal curve from \( Z_1 \) to \( Z_2 \) = (normal-area \( Z_1 \) \( Z_2 \)) = .2632 or 26.32 percent
   Of course, you may find the approximate normal area from the normal table in the book:
   Area (1.75) = 91.99  (area between 19.5 and 36.5)
   Area (0.50) = 38.29  (area between 25.5 and 30.5)
   Difference = 53.70  (gives the two areas from 19.5 to 25.5 and 30.5 to 36.5)
   Half the difference = 26.85 (since we only want half the area)
Chapter 18, problem 1. Finding a normal area using the text tables:

The following code may help you see the text method of finding this area:
If you have closed the histogram, redraw it:

(hist sums integers)

Then shade in the two areas we found above, in different colors:

(shadebins 19.5 36.5 red )

Why to 36.5? The text table will give the normal area from -1.75 standard error units to 1.75 SE units. This is from 19.5 - 28 = -8.5 or 8.5 units below the mean of 28 to 8.5 units above the mean = 28 + 8.5 = 36.5

(shadebins 25.5 30.5 magenta)

The difference of the two gives the two red areas, of which we are only interested in one.
Problem 2. Four hundred draws with replacement from the box: [1, 3, 5, 7]
Mean of box = 4; mean squared deviation = 5; SD = (sqrt 5) = 2.2361
EV of sum of 400 draws = 1600  SE of sum of 400 draws (sqrt 400) * (sqrt 5) = 44.7214

a. Chance that sum of 400 draws is greater than 1500.
   \[ Z = \frac{1500 - 1600}{44.7214} = -100/44.72 = -2.2361 \]
   \[ (\text{pnorm} -2.2361) = 0.0127 \text{ or } 1.27 \text{ percent chance of a value LESS THAN 1500,} \]
   Hence there is a 98.73 chance of a value greater than 1500

a1. Chance that the sum of 400 draws is between 1650 and 1700
   \[ Z_1 = \frac{1650 - 1600}{44.7214} = 50/44.7214 = 1.1180 \]
   \[ Z_2 = \frac{1700 - 1600}{44.7214} = 100/44.7214 = 2.2361 \]
   (normal-area z1 z2) = .1191 or 11.91 percent
   Text method using normal tables:
   Area (Z2) = Area (2.25) = 97.56
   Area (Z1) = Area(1.10) = 72.87
   Difference = 24.69, and half the difference = 12.345 is our (rough) approximation.

b. Chance that there are fewer than 90 threes in 400 draws:
   First, change the box to reflect "getting a 3" as the desired outcome: box = [0, 1, 0, 0]
   Mean of box = 0.25; SD of box = (sqrt (.25 .75)) = 0.4330
   EV of sum of 400 draws = 100 * .25 = 100
   SE of sum of 400 draws from this box = (sqrt 400) * 0.4330 = 8.6603
   Read "fewer than 90" as less than 85.5, so the z-score of the critical value is:
   \[ Z = \frac{89.5 - 100}{8.6603} = -10.5/8.6603 = -1.2124 \]
   \[ (\text{pnorm} -1.2124) = 0.1127 = 11.27 \text{ percent gives the value directly and exactly.} \]
   Text method:
   \[ 100 - (\text{area 1.20}) = 100 - 76.99 = 23.01 \text{ gives the two tail area;} \]
   the desired area would be half this value or 11.505 percent.

Simulate this box with the commands:
   (bind box (list 0 1 0 0))   (bind sums nil)
   (dotimes (i 10000) (push (sum (draw 400 box)) sums))
   (hist box integers)
   (shadebins 89.5 110.5 blue)
   We want to find the left tail area.
   (prob (< sums 90)) will tell us the percentage of the simulated values in that bin.

Problem 3. Ten draws with replacement are made from the box [0, 1, 2, 3]. Chance that sum is in the interval from 10 to 20 is equal to the area under the probability histogram from 9.5 to 20.5. It is approximated by the area under the normal curve which can be found by first calculating the mean and SD of the box: Mean of box = 1.5; SD of box = (sqrt 1.25) = 1.1180 and then by finding the expected value of the sum of the 10 draws = 10 * 1.5 = 15 and the SE of the sum = (sqrt 10) * (sqrt 1.25) = 3.5355. The z-scores of the critical values of 9.5 and 20.5 are +/- (20.5 - 15) / 3.5355 = +/- 1.5556, so the appropriate area under the normal curve is (from text tables) 87.89 %, or from the EcLS command (normal-area -1.5556 1.5556) = .8802 or 88.02 %
Simulation of this problem as in the previous problem (with of course the new box and the command (shadebins 9.5 20.5 green) should result in a close approximation (My run yielded 88.10 % of the data within those bins).
Problem 4. A coin is tossed 25 times. What is the chance of getting exactly 12 heads and 13 tails. With the unbiased coin assumed in the problem, this probability will be $0.1550$ or $15.50\%$. To make the procedure clearer, I will suppose that the coin is unfair, with a 60 percent chance of heads. This would yield the following binomial computation, in EcLS code, with $X =$ number of heads in 25 tosses:

I first issued the command (decimals 8) to allow the display of $(\text{pow} \, 0.4 \, 13)$

$$\text{Pr} \,(X = 12) = (\ast \, (\text{combinations} \, 25 \, 12) \, (\text{pow} \, .6 \, 12) \, (\text{pow} \, .4 \, 13))$$
$$= (\ast \, 5,200,300 \quad 0.00217678 \quad .00000671)$$
$$= 0.07596671$$
$$= 7.60\%$$

A complete printout of all probabilities for this binomial distribution is available by the command:

(binomial-dist 25 .6)

A more direct calculation would be (binomial-pmf 12 25 0.6) = (dbinom 12 25 .6) = 0.07596671.

The first command is the original EcLS version, with "pmf" standing for "probability mass function", the analog of the density function for discrete rather than continuous distributions; the "dbinom" is the S or R version, which follows the pattern of using a "d" for density function even for continuous distributions.

Problem 5. Twenty-five draws with replacement from box = \{1, 1, 2, 2, 3\}.

Construct a histogram from the numbers drawn, for the probability histogram of the sum, and for the probability histogram of the product. Identification will be easy with the following simulation:

(bind box (list 1 1 2 2 3))
(bind draws (draw 25 box)) (hist draws integers) -- this is for one actual draw;
whether the bar representing 1 or 2 or even possibly 3 is highest depends on pure chance.

(bind sums nil) (bind products nil)
(dotimes (i 1000) (bind newdraw (draw 25 box))
   (push (apply #'+ newdraw) sums) ;; could also say (sum newdraw)
   (push (apply #'* newdraw) products) ;; can't yet say (product newdraw), will be able to in
) ;; EcLS 9.01 and higher.

(hist sums integers) ;; asking for "integers" on the products may well freeze up your system's memory, ;; and remind you that Ctrl-Alt-Delete is sometimes necessary to kill the process ;; look for wxls32.exe in the Processes tab of the Task Manager.

Problem 6. The COIN program to simulate coin tosses is tested by running 1,000,000 simulations of a coin toss.

Try this on EcLS = (set! tosses (round (runif 1000000)))
I use an alternative to "bind" -- set! does not print out the million numbers it generates.
You may use (sample tosses 100) to view a subsample of the tosses.
It generates 1,000,000 random uniform numbers on the interval 0 to 1, and rounding of course translates this into zero if below 0.5 and 1 if above.
(sum tosses) would have a value of 500,000 if there were no chance error at all -- but of course you will get some chance error in any single set of random numbers.

In the text example, the random error was 502,015 -- was your run that far off?
The text programmer correctly notes that the percentage error is more relevant than the absolute error -- but how much of a percentage error should we expect? What is the SE of the sum of a million coin tosses?

Mean and SD of the box [0,1]:  Mean = 0.5, SD = 0.5
Mean of a million tosses = 500,000  SE of a million tosses = (sqrt 1000000) * 0.5 = 500, so 2,015 tosses off is more than 4 standard error units off -- and this is very improbable if the program is working correctly.