The following numbered questions should be split across your group and the solutions discussed during the next lecture period. Students should review the learning goals for the day, determine which are applicable to their questions and provide answers or commentary to their group members. When using the Internet to formulate answers (some questions may require this), keep track of **where** you find your information on the web. You may be asked for, and are expected to have (in Email-able form), URLs supporting your investigations.

The questions below refer to specific data files and an `sis1` program. These can be downloaded `here` or from the course schedule page just like the last LGA.

Download the tarball, unravel it, and check the `README.txt`

```
workdir$ tar xjf coding-sis.tar.bz2
workdir$ cd coding-sis
workdir$ more README.txt
```

in your preferred working directory.

For all programming questions, you will need to modify the provided code to report actual dollar values associated with your investigations. You should be prepared to walk your group through the changes made to the given baseline code for your particular question, as well as the results generated.

1. Question 1.3.4 (§1.3.6) The book's errata corrects this question: it should be Figure 1.3.9. You are not expected to generate a four-panel figure, just generate the four graphs (for each S) so that your group may discuss the results when you meet again.

2. Similar to question 1, investigate how the optimal s changes as the setup and holding costs increase by factors of $1.10, 1.20, 1.40$. **That's nine experiments**, not three — for each value of setup cost, you'll run your analysis for all three possible values of the holding cost.

3. Question 1.3.8 (§1.3.6) Clarification: a circular array is simply a conventional array of data (length L) but with an arbitrary starting point ($0 \leq sp < L$) and an accessor method (or convention) so when **index** x is read the value at **index**

$$(sp + x) \bmod L$$

   is actually used. Calculate the optimal s for the car lot for each ordering of demands and present your results in either tabular or graphical form (or both!). As in 1.3.9, consider **at least** 10 different starting points.

4. Question 1.3.9 (§1.3.6) Even though we haven't considered pRNGs and shuffling algorithms in the course, you are welcome to use your preferred language's builtin support for randomizing arrays. Alternatively, the Linux OS the `sort(1)` and `shuf(1)` commands that will do the job for you at the command line. Consider for example:

```
workdir$ ./sis-cxx <(shuf sis1.dat)
```

   Calculate the optimal s for the car lot for each shuffling of demands and present your results in either tabular or graphical form (or both!). Consider **at least** 10 different starting points (but don't consider all possible orderings `:)`).