ENGR 3520: Foundations of Computer Science
Professor Lynn Andrea Stein
Fall 2010

Assignment 1 FAQ

Coding Problems, Generally

Q. I'm assuming it's OK to test out the procedures we write in DrRacket for debugging?

A. Yes, when producing code, please do test and then copy/paste rather than retyping with errors. However, problem 3 is not a coding problem in this sense. See problem 3 FAQ below.

Q. Can we use built in scheme functions that we have not talked about in class? Like append?

A. In general, yes. I had intended for this pset to be done without append, but I will not ding those who do it (correctly). So it is neither necessary nor explicitly forbidden....

Problem 1

Q. 1 c. states that we need to make a "procedure that takes one argument, a positive integer, and returns a list of all of the integers between 1 and that number, inclusive."

I was wondering if, when defining the function, we can have an inner define statement? I don't think it's possible to do this tail-recursively with just one argument (unless we could make that 1 argument a list, which is against the guidelines). We haven't done inner functions in class, so I just wanted to double check.

A. You are welcome to define an inner function. In scheme, any time an inner function would work, a top level function would work as well (though it exposes the name to the global namespace). So you can definitely use a helper function and you can define it inside or outside as you prefer.

Q. For the challenge part on question one, is the function allowed to take in more than 1 argument?

A. Yes, although it is possible to do it both ways using only one argument in each case. (In one case you need either an extra argument or a helper function, but it is actually possible to use a helper function with only one argument. Don't worry about this, though; just write the function however you can.)

Q. Does my procedure need to check that the argument is correct (a positive integer)?

A. No. You can assume that the argument will be in the correct form; I want you to focus on the key idea and not the software engineering for this problem set.

Problem 2

Q. For question 2 can the procedure take in more than 1 argument?

A. The procedure described can be written without taking in more than one argument, but you may use a helper procedure that does. If you can only make it work with more than 1 argument, you should write it that way, of course.

Q. I am having some trouble with ... #2. <Code details omitted here> ...I have been struggling with this question for a few hours, and I feel that I there is something I don't understand

A. A few things:

1. Good to recognize that you're wedged. More head-pounding isn't likely to unwedge you, and seeking some redirection is a Good Idea.

2. I am guessing that at some point in your English description, there's a point at which you want to CHANGE something rather than creating and using a new, similar-but-slightly-dfferent thing. If that's true, you're not going to be able to write this code without using assignment, and you're not supposed to use assignment on this problem. (One sign of this is when you refer back to "the list" and mean something different from what you were originally handed....)

3. If I'm right about #2, you shouldn't be writing code (yet). You should be reworking your English description until you have a version that doesn't involve changing anything. (Your description would then say "the new list" instead of "the list"....)

4. **This is definitely a hard question.**

Problem 3

Q. Are we allowed to use DrRacket for the problem set? More specifically, when we are asked in problem 3 what the results of different expressions involving “repeatedly,” do you want us to answer based on what we expect to happen, or what happens when we actually enter the expressions in DrRacket?

A. I would like you to (try to) figure it out. You may use the environment to double check. If you change your answer, please indicate this and explain the difference, i.e., what actually happens that you didn't expect.

In other words, this isn't supposed to be a test as to whether you can run code successfully; it's supposed to be you thinking about how it works. But you can use the environment to see and learn once you've thought it through.

Q. For question 3, what do you want us to explain for part f? Should we just talk through why parts a-e gave us those answers? Or why some parts gave us different answers and other ones gave us the same?

A. If you do the first thing you say, you will have done the second as well, I think.

Problem 4

Q. Problem 4 reads:

Read about (stacks and) queues in Cormen (pp. 200 - 203).

Cormen 3rd ed. has a description of bucket sort on pp. 200-203. They describe stacks and queues on pp. 232-236. I suspect this is a typo.

A. Actually, I think it is version skew. At least one version of Cormen (the one on google books, which is the only one I have within reach at the moment to be able to double-check) seems to think that section 10.1 Stacks and Queues begins on p.200. However, you should exercise responsible judgement with respect to page numbers, especially if you are using an algorithms book other than Cormen or an edition other than the (severely dented) one I used when making up the syllabus. It is more important that you look up queues than that you reference pages 200-203, specifically 

Problem 5

Q. For question 5, it says that we can use the expression evaluators discussed in class... can I get the source code for that?

A. It was linked from the front page of the wiki, but it's now linked from problem 5 as well. It is at ExpressionEvaluatorDotScm.