In Exercise 6,

"You are to determine the final average using treeflatten, map, and accumulate, exclusively, with no other user-defined functions."

1. I am not sure that accumulate function is predefined in Scam?

2. Are we allowed to use predefined function - (length list) in Exercise 6?

Accumulate is not predefined, so you must define it (it is defined for you in the textbook).

You can use predefined functions.

Can treeflatten have a nested helper?

Yes.

So the arguments to map and accumulate have to be built-in functions?
You can use simple lambdas that reference built-ins for map and accumulate.

The restriction is put in place so you don’t write a special purpose function to calculate the answer.

Alright. Would it be okay to define some additional helper functions outside of task 6 like:

getValue
getRight
getLeft
IsNull?

Those would be fine.

Remember that these built in functions exist:

nil?
car
cadr
caddr

Should the result of tree flatten be a list of pairs such as ( (1.2) (2.3) (3.4) )
That is correct, with the depth coming first.

I've just been using the built-in getElement function, should I be using the car, cadr, and caddr functions instead?

I think it would be more preferable to use car, cadr, and caddr; but I don't see any issue with using getElement unless Dr. Lusth says not to.

Is there a built-in trace function to observe my function while it's running?

Just use println and inspect. No built in trace functions.

You can trace an error, though, with -t N, where the larger the N, the more tracing information.