Exercise 1

1. The two lines of code responsible for the infinite scrolling are:  
     
   If we were to remove those lines, the list would end after the first 10 pairs, and we wouldn’t be able to keep on scrolling. That was my guess, at least. But we get a range error, because we try to read values from outside the lists range.
2. We could use a separator builder instead, and then instead of building items and dividers by separating even and odd indexes, we have a different builder for items and for separators. Personally, I think this way is better because it makes more sense to have a real separation between the list items and their separators. In other words, the separators are not a part of the list’s items, so they shouldn’t be built in the same place as them.
3. We need a call to setState() to tell the app to build its UI again. If we just update the saved set, but don’t actually build the UI, we won’t see the change.

Exercise 2

1. I used Navigator.of(context).push() to push a new MaterialPageRoute into the navigator stack. Another way to do the same thang is to define two statelessWidgets – screens/routes from the get-go and then defining routes in the MaterialApp constructor and use Navigator.pushNamed() with the route’s name defined in the constructor.
2. I used ScaffoldMessenger.of(context).showSnackBar(). A Scaffold Widget is required to show a snackbar, because the ScaffoldMessenger actually shows a snackbar in all the scaffolds of the app simultaneously, so the message doesn’t disappear when moving through different screens. No scaffolds – no snackbars.