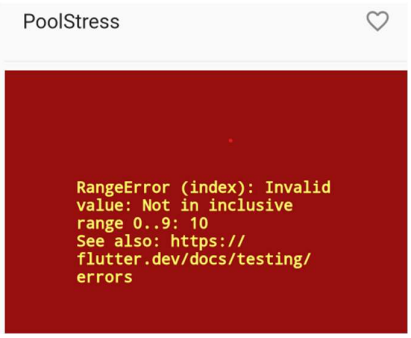
Dry Part

Exercise 1

1. The lines are:

```
if (index >= _suggestions.length) {
  _suggestions.addAll(generateWordPairs().take(10)); // We can comment only
this line of course...
}
```

   When commenting those lines, we actually not adding further lines to the original 10 that were declared in the constructor and therefore an error will be occurred when scrolling and index is bigger than 10:



2. As described on the question, we are dealing with finite list, so we can use the "ListView.separated" constructor that will take care of the Divider items in a separate constructor. The answer to the question what option is the best mainly depend on what we want. For absolutely most cases we definitely want to use the "ListView.separated" constructor, we don't what to handle the creation of the Divider manually and check the index position like we did in the assignment, also it "pollutes" the role and logic of the "itemBuilder" which suppose to handle only the building of the item itself, and the Divider is definitely another item which causing clumsy calculations of the item's indexes. But, it's possible in minor cases that we want to control over the Divider functionality ourselves, for example creating Divider not between every item pair but rather under another logic followed by conditions etc.
3. We want the change of the _saved property (remove/add) will effects the "RandomWordsState" Widget. Therefore, we need the Widget build method will be called by the framework, this is what setState does. The callback for remove/add an item of the _saved property is called synchronously and as a result the framework re-runs the build method.

Exercise 2

1. The purpose of MaterialApp widget is to make and implement components that are follow the guidelines of Material Design ,which support the nest practices of user interface design, regardless of the platform the users are using.

   Property num 1: color.

With MaterialApp we define a primary color theme which helps craft a consistent color that can be used globally throughout the application and helps users perceive content and easily know how the app works

Property Num 2: supportedLocales

The list of locales that this app has been localized for. By default, only English locale is supported. Apps should configure this list to match the locales they support.

Property Num 3: title

A one-line description used by the device to identify the app for the user (This is not the title which is displayed on the main screen). For example, on Android devices the titles appear above the task manager's app snapshots which are displayed when the user presses the "recent apps" button.

2. The “Key” property distinguishes every item on the “Dismissable” list from other items and must be unique so the “onDismissed” callback will be applied only on a specific item. The “Key” property must be required because without them, the default behavior is to sync widgets based on their index in the list, which means the item after the dismissed item would be synced with the state of the dismissed item. Using keys causes the widgets to sync according to their keys and avoids this pitfall.