תרגיל בית 1:

Dry exercise 1:

Q1: these 2 lines:

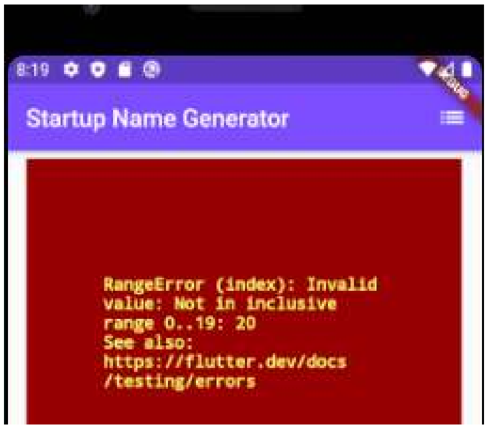
```
if (index >= _suggestions.length) {
  _suggestions.addAll(generateWordPairs().take(10));

}
```

These lines make the scrolling infinite..because everytime I reach the end of the list, the listview builder will insert 10 more wordpairs to the" _suggestions[index] " array and then build them i.e insert them into the listview builder

Now If we remove these lines and assume we already have 20 rows in the list..what will happen is that we will get a range error when we reach the end of the list because in our _buildSuggestions widget we have the following line in the end : **return _buildRow(_suggestions[index]);**

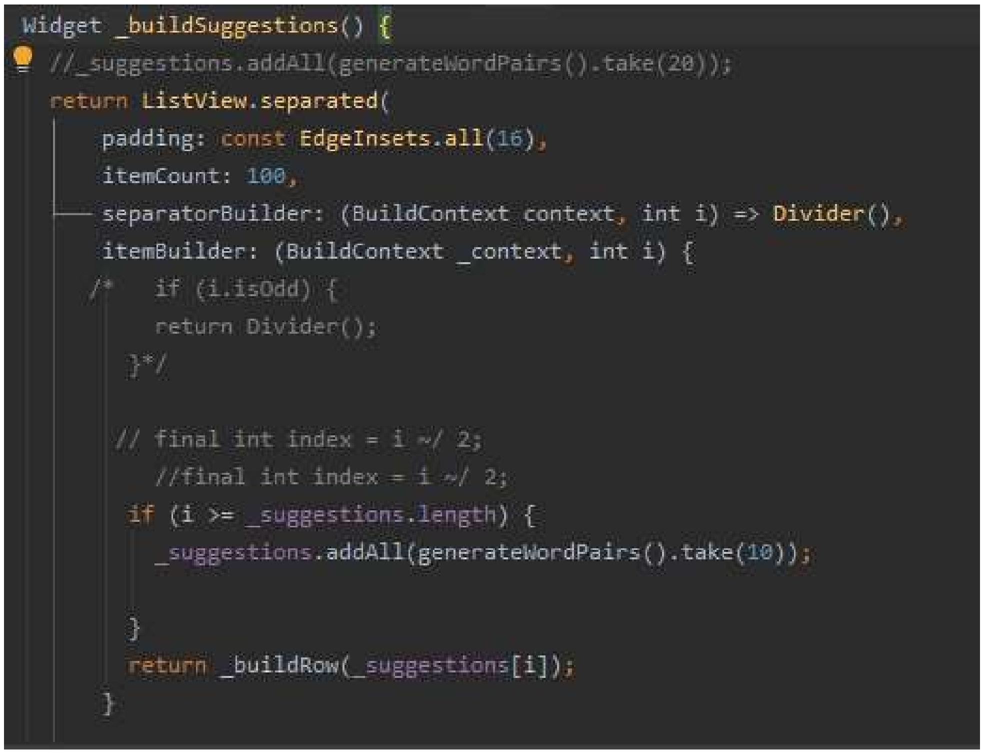
This line calls _buildRow function which builds rows and puts in each row the wordpair in _suggestions[index],but if we remove the lines I mentioned above what will happen is that the index will keep growing and we will access a location in the array that does not contain any valid value inside it and so we will receive the following error



```
RangeError (index): Invalid
value: Not in inclusive
range 0..19: 20
See also:
https://flutter.dev/docs
/testing/errors
```

Q2: Yes,if we assume that the listview is finite and has 100 items then we can build it using ListView.Seperated instead of ListView.Builder

The new code will look like this



```
Widget _buildSuggestions() {
  //_suggestions.addAll(generateWordPairs().take(20));
  return ListView.separated(
      padding: const EdgeInsets.all(16),
      itemCount: 100,
      separatorBuilder: (BuildContext context, int i) => Divider(),
      itemBuilder: (BuildContext _context, int i) {
     /*   if (i.isOdd) {
          return Divider();
        }*/

       // final int index = i ~/ 2;
          //final int index = i ~/ 2;
        if (i >= _suggestions.length) {
          _suggestions.addAll(generateWordPairs().take(10));

        }
        return _buildRow(_suggestions[i]);
      }
```

Now this way may look better than the original code but to use listview. separated I need to have a finite list which is a disadvantage (it means I have a limit on the lists size or that I need to know the size of my list that im building).

Q3:

When we tap the heart icon, we first of all want the heart icon to change colors,if its red we want it to become transparent and if its transparent we want it to become red, now our main widget(RandomWords) is a stateful widget and we need to notify it that we tapped on the heart (we need to notify it that "the state has changed"), so we need to use the setstate() call and it will rebuild the widget and we will get an updated widget..if we remove setstate() the heart icon wont change,and in turn ofcourse if we access the favorites page (the list icon on the right up) it might not be updated.

Dry 2 – Exercise 2:

Q1: we used Navaigator.of(context).push

and we created a new MaterialPage route

another way to do this would be to navigate using named routes method, ie `Navigator.pushNamed()` method

Q2: I created a global Scaffold key,assigned this key to be the scaffold that represents the page im currently in,and then I used the showSnackBar method,I can only show a snackbar within a scaffold..

The method used was:

```
_scaffoldKey.currentState.showSnackBar(SnackBar
( content:  Text("Login is not implemented
yet")));
```

another way to show a snackbar could be done like this:

```
Scaffold.of(context).showSnackBar(snackBar);
```

But this requires us to have the context of the scaffold so we can show the snackbar..by using a global key like before I only need to assign the key to the scaffold I want..i do not need the context when calling showSnackBar