NAME: SAKSHI VINAYAK KITTURKAR
CSU ID: 2860273
HOMEWORK 3

Develop a notepad app with custom keypad:

- Top fragment: display the note entered.
- Bottom fragment: display several rows of letters, numbers, and symbols (each as a button) for text input; when a user pushes a button, the corresponding input is appended in the top fragment.

= Developed a notepad with two fragment at the top and the bottom the top one will display the notes entered and the bottom fragment has rows of letters, numbers, symbols from where we give input.

In layout: Activity_One.java this code represents the definition of an Android activity (Activity_One) and specifies its layout using an XML file named "activity_one.xml." This activity is part of an Android app in the "com.example.hw3" package. The actual layout and functionality of the activity would be defined in the corresponding XML file and any additional Java code associated with it.

Activity_One that extends AppCompatActivity. The onCreate method is crucial in Android activities and is responsible for initializing the activity. In this case, it sets the content view of the activity using the layout defined in the XML file named "activity_one.xml

```
public class Activity_One extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_one);
    }
}
```

In layout: Fragment1.java this code illustrates a typical structure for an Android Fragment, incorporating elements for parameter handling, layout inflation, and interaction with a ViewModel for data management. The fragment is designed to display information in a TextView based on data changes observed from the associated ViewModel.

Class Declaration

```
public class Fragment1 extends Fragment {
```

Static Factory Method

```
public static Fragment1 newInstance(String param1, String param2) {
   // ...
}
```

Instance Variables

```
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";

private String param1;
private String param2;
```

Constructor and onCreate method

```
public Fragment1() {
   // Required empty public constructor
}

@Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   // ...
}
```

onCreateView method

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
   // Inflate the layout for this fragment
   return inflater.inflate(R.layout.fragment_1, container, false);
}
```

ViewModel Interaction

```
MyViewModel viewModel = new
ViewModelProvider(requireActivity()).get(MyViewModel.class);
```

```
TextView textViewTyped = view.findViewById(R.id.textview);

viewModel.getData().observe(requireActivity(), typedString -> {
    textViewTyped.setText(typedString.toString());
});
```

In layout: Fragment2.java this code named Fragment2 is used for functionality for handling arguments, interacting with a ViewModel.

Declares a class named Fragment2 that extends the Fragment class.

```
public class Fragment2 extends Fragment {
```

Provides a static factory method (newInstance) to create an instance of Fragment2 with specified parameters.

```
public static Fragment2 newInstance(String param1, String param2) {
    // ...
}
```

Defines constant strings for argument keys (ARG_PARAM1 and ARG_PARAM2), instance variables to store the fragment's parameters, and a MyViewModel instance for interaction with a ViewModel.

```
private static final String ARG_PARAM1 = "param1";
private static final String ARG_PARAM2 = "param2";

private String param1;
private String param2;
private MyViewModel viewModel;
```

Provides a default constructor and overrides the onCreate method to handle any initialization, such as retrieving arguments.

```
public Fragment2() {
    // Required empty public constructor
}
```

```
@Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   // ...
}
```

Overrides the onCreateView method to inflate the fragment's layout from the XML file "fragment_2.xml."

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
   // Inflate the layout for this fragment
   return inflater.inflate(R.layout.fragment_2, container, false);
}
```

Overrides the onViewCreated method to perform actions after the fragment's view has been created. It initializes the ViewModel and sets up event listeners for user interactions.

```
@Override
public void onViewCreated(View view, Bundle savedInstanceState) {
   super.onViewCreated(view, savedInstanceState);
   // ...
}
```

In layout: MyViewModel.java This code defines an Android ViewModel class named MyViewModel that is responsible for managing and storing data for use by associated UI components, such as fragments or activities.

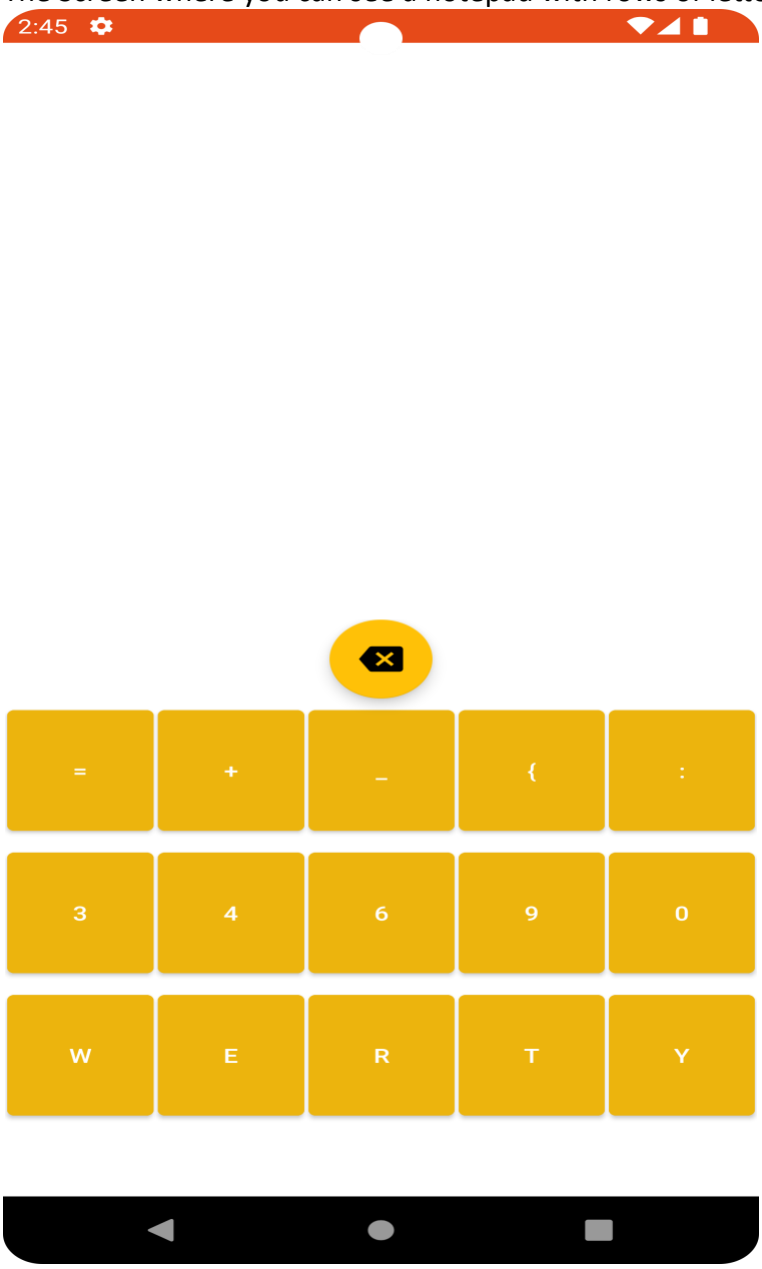
Provides a method to increment the stored data by appending the character a to the current value. This triggers observers (such as UI components) to be notified of the data change.

```
public void incrementData(char a) {
   data.setValue(data.getValue() + a);
}
```

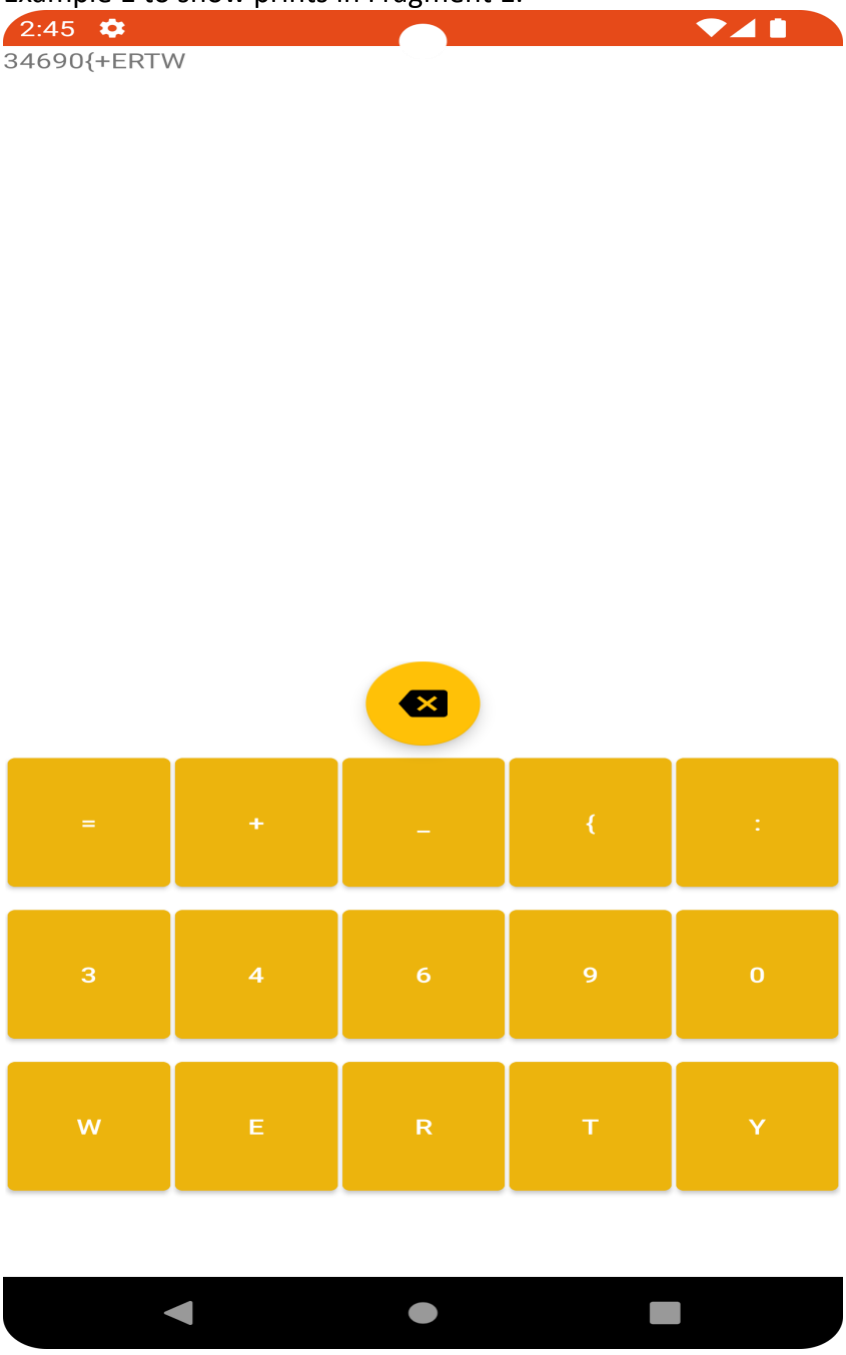
Provides a method to decrement the stored data by removing the last character from the current value. This method ensures that the length of the data remains greater than zero before performing the operation.

```
public void decrementData() {
   if (data.getValue() != null && data.getValue().length() > 0) {
      data.setValue(data.getValue().substring(0, data.getValue().length() - 1));
   }
}
```

The Screen where you can see a notepad with rows of letters, number, and symbols.



Example 1 to show prints in Fragment 1.



Example 2 to show prints in Fragment1.