

Q.2] Create an Android application which will ask the user to input his / her name. A message should display the two items concatenated in a label. Change the format of the label using radio buttons and check boxes for selection. The user can make the label text bold, underlined or italic as well as change its color. Also include buttons to display the message in the label, clear the text boxes as well as label. Finally exit

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    tools:context=".MainActivity">
```

```
    <EditText
```

```
        android:id="@+id/editTextName"
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:hint="Enter your name"
```

```
        android:layout_marginTop="16dp"
```

```
        android:layout_marginStart="16dp"
```

```
        android:layout_marginEnd="16dp"/>
```

```
    <!-- RadioGroup for text format selection -->
```

```
    <RadioGroup
```

```
        android:id="@+id/radioGroupFormat"
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
android:layout_below="@id/editTextName"
android:orientation="horizontal">
```

```
<RadioButton
    android:id="@+id/radioButtonBold"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bold"/>
```

```
<RadioButton
    android:id="@+id/radioButtonItalic"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Italic"/>
```

```
<RadioButton
    android:id="@+id/radioButtonUnderline"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Underline"/>
```

```
</RadioGroup>
```

```
<!-- CheckBoxes for text style selection -->
```

```
<CheckBox
    android:id="@+id/checkboxColor"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
android:text="Change Color"
android:layout_below="@id/radioGroupFormat"
android:layout_marginTop="16dp"
android:layout_marginStart="16dp"/>
```

<CheckBox

```
android:id="@+id/checkBoxClear"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Clear Text"
android:layout_below="@id/checkBoxColor"
android:layout_marginTop="16dp"
android:layout_marginStart="16dp"/>
```

<Button

```
android:id="@+id/buttonDisplay"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Display Message"
android:layout_below="@id/checkBoxClear"
android:layout_marginTop="16dp"
android:layout_marginStart="16dp"/>
```

<Button

```
android:id="@+id/buttonExit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
android:text="Exit"
android:layout_below="@id/buttonDisplay"
android:layout_marginTop="16dp"
android:layout_marginStart="16dp"/>
```

```
<TextView
```

```
    android:id="@+id/textViewMessage"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="18sp"
    android:textStyle="normal"
    android:textColor="@android:color/black"
    android:layout_below="@id/buttonExit"
    android:layout_marginTop="32dp"
    android:layout_marginStart="16dp"/>
```

```
</RelativeLayout>
```

```
MainActivity.java:
```

```
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.RadioButton;
```

```
import android.widget.RadioGroup;

import android.widget.TextView;


import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity {

    private EditText editTextName;

    private RadioGroup radioGroupFormat;

    private CheckBox checkBoxColor, checkBoxClear;

    private Button buttonDisplay, buttonExit;

    private TextView textViewMessage;


    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);


        editTextName = findViewById(R.id.editTextName);

        radioGroupFormat = findViewById(R.id.radioGroupFormat);

        checkBoxColor = findViewById(R.id.checkBoxColor);

        checkBoxClear = findViewById(R.id.checkBoxClear);

        buttonDisplay = findViewById(R.id.buttonDisplay);

        buttonExit = findViewById(R.id.buttonExit);

        textViewMessage = findViewById(R.id.textViewMessage);


        buttonDisplay.setOnClickListener(new View.OnClickListener() {

            @Override
```

```
public void onClick(View v) {  
    displayMessage();  
}  
});
```

```
buttonExit.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        finish();  
    }  
});  
}
```

```
private void displayMessage() {  
    String name = editTextName.getText().toString().trim();  
    StringBuilder message = new StringBuilder();  
  
    if (!name.isEmpty()) {  
        message.append(name);  
    }  
}
```

```
int selectedFormatId = radioGroupFormat.getCheckedRadioButtonId();  
RadioButton selectedFormat = findViewById(selectedFormatId);  
int style = TextView.NORMAL;  
  
if (selectedFormat != null) {  
    switch (selectedFormat.getId()) {
```

```

        case R.id.radioButtonBold:
            style |= TextView.BOLD;
            break;

        case R.id.radioButtonItalic:
            style |= TextView.ITALIC;
            break;

        case R.id.radioButtonUnderline:
            // Underline is not supported directly by TextView, handled through spans
            // You can implement underline functionality here
            break;
    }
}

if (checkBoxColor.isChecked()) {
    textViewMessage.setTextColor(Color.RED);
} else {
    textViewMessage.setTextColor(Color.BLACK);
}

textViewMessage.setText(message.toString());
textViewMessage.setTypeface(null, style);

if (checkBoxClear.isChecked()) {
    editTextName.setText("");
    textViewMessage.setText("");
}
}

```

}