Program 47 : Write a program that takes two CheckBox ‘Bold’, ‘Italic’ and one TextView having text "Saurashtra University by clicking on CheckBox apply selected effect on TextView. (Use setTypeface())

- Activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <CheckBox
        android:id="@+id/b"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <CheckBox
        android:id="@+id/i"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <TextView
        android:id="@+id/t"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Suarastra University"
        android:textSize="20sp"/>
</LinearLayout>
```

- MainActivity.kt

```kotlin
import android.graphics.typeface
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
       val f = {
        t.setTypeface(null,
        if (b.isChecked && i.isChecked){
            typeface.BOLD_ITALIC
        }else if (b.isChecked){
            typeface.BOLD
       }else if (i.isChecked){
            typeface.ITALIC
        }else{
            typeface.NORMAL
        })
       }
        b.setOnClickListener { f() }
        i.setOnClickListener { f() }
    }
}
```

Program 48,49,50 : This Android application demonstrates the use of ListView and Spinner with ArrayAdapter. Cities are stored in string.xml and fetched using getResources() and getStringArray() to display in a ListView. On selecting an item, the selected value is shown using Toast. It also displays color options (Red, Green, Blue) in a ListView, and when a color is selected, the background color of the screen changes using setBackgroundColor().

- String.xml

```
<resources>
    <string-array name="cities">
        <item>Rajkot</item>
        <item>Ahmedabad</item>
        <item>Surat</item>
    </string-array>

    <string-array name="colors">
        <item>Red</item>
        <item>Green</item>
        <item>Blue</item>
    </string-array>
</resources>
```

- Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

- MainActivity.kt

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val listView = findViewById<ListView>(R.id.listView)
        val spinner = findViewById<Spinner>(R.id.spinner)
        val layout = findViewById<LinearLayout>(R.id.layout)
        val cities = resources.getStringArray(R.array.cities)
        val colors = resources.getStringArray(R.array.colors)
        val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
     cities)
        listView.adapter = adapter
        spinner.adapter = ArrayAdapter(this,
               android.R.layout.simple_spinner_dropdown_item, cities)
        listView.setOnItemClickListener { _, _, pos, _ ->
            Toast.makeText(this, cities[pos], Toast.LENGTH_SHORT).show()
            when (colors[pos % 3]) {
                "Red" -> layout.setBackgroundColor(Color.RED)
                "Green" -> layout.setBackgroundColor(Color.GREEN)
                "Blue" -> layout.setBackgroundColor(Color.BLUE)
            }
        }
        spinner.onItemSelectedListener = object :
           AdapterView.OnItemSelectedListener {
                override fun onItemSelected(p: AdapterView<*>, v: View?, pos:
                     Int, id: Long) {
                          Toast.makeText(this@MainActivity, cities[pos],
                          Toast.LENGTH_SHORT).show()
                }
                override fun onNothingSelected(p: AdapterView<*>) {}
           }
     }
}
```

Program 51 : Write a program that when a button is clicked, the ProgressBar will begin to progress until it reaches 100. (Use Horizontal ProgressBar, Thread)

- Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ProgressBar
        android:id="@+id/progressBar"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:max="100"/>
    <Button
        android:id="@+id/btnStart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start"/>
</LinearLayout>
```

- MainActivity.kt

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val pb = findViewById<ProgressBar>(R.id.progressBar)
        val btn = findViewById<Button>(R.id.btnStart)
        btn.setOnClickListener {
            Thread {
                for (i in 0..100) {
                    pb.progress = i
                    Thread.sleep(50)
                }
            }.start()
        }
    }
}
```

Program 52, 53 : This Android application uses AutoCompleteTextView and MultiAutoCompleteTextView. City names are stored in strings.xml and fetched using getStringArray(). Suggestions are provided using ArrayAdapter and setAdapter(). AutoCompleteTextView shows single suggestions, while MultiAutoCompleteTextView provides multiple suggestions using setTokenizer().

- String.xml

```xml
<resources>
    <string-array name="cities">
        <item>Rajkot</item>
        <item>Ahmedabad</item>
        <item>Surat</item>
        <item>Vadodara</item>
    </string-array>
</resources>
```

- activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <AutoCompleteTextView
        android:id="@+id/autoCity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter City"/>
    <MultiAutoCompleteTextView
        android:id="@+id/multiCity"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Multiple Cities"/>
</LinearLayout>
```

- MainActivity.kt

```kotlin
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        val cities = resources.getStringArray(R.array.cities)
        val adapter = ArrayAdapter(this,
            android.R.layout.simple_dropdown_item_1line, cities)
        findViewById<AutoCompleteTextView>(R.id.autoCity)
            .setAdapter(adapter)
        findViewById<MultiAutoCompleteTextView>(R.id.multiCity).apply {
            setAdapter(adapter)
            setTokenizer(MultiAutoCompleteTextView.CommaTokenizer())
        }
    }
}
```

Program 54,55 : This program uses DatePickerDialog and TimePickerDialog in Android. The user selects a date and time using dialog boxes. After selection, the chosen date and time are displayed in a TextView. The application is developed using Kotlin.

- Activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/txtResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Selected Date - Time"/>

    <Button
        android:id="@+id/btnDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Date"/>

    <Button
        android:id="@+id/btnTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select Time"/>
</LinearLayout>
```

- MainActivity.kt

```kotlin
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val txt = findViewById<TextView>(R.id.txtResult)

        findViewById<Button>(R.id.btnDate).setOnClickListener {
            val c = Calendar.getInstance()
            DatePickerDialog(this,
                { _, y, m, d -> txt.text = "Date: $d/${m+1}/$y" },
                c.get(Calendar.YEAR),
                c.get(Calendar.MONTH),
                c.get(Calendar.DAY_OF_MONTH)
            ).show()
        }

        findViewById<Button>(R.id.btnTime).setOnClickListener {
            val c = Calendar.getInstance()
            TimePickerDialog(this,
                { _, h, min -> txt.text = "Time: $h:$min" },
                c.get(Calendar.HOUR_OF_DAY),
                c.get(Calendar.MINUTE),
                true
            ).show()
        }
    }
}
```

Program 56, 57, 58 : This Android application demonstrates the use of AlertDialog with different buttons. The dialog shows Ok, Yes–No, and Ok–Cancel options. When the user clicks any button, the corresponding message is displayed in a TextView. The dialog buttons are implemented using BUTTON_NEUTRAL, setPositiveButton(), and setNegativeButton() in Kotlin.

- Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/txtResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Result"/>

    <Button
        android:id="@+id/btnOk"
        android:text="Alert OK"/>

    <Button
        android:id="@+id/btnYesNo"
        android:text="Alert Yes No"/>
    <Button
        android:id="@+id/btnOkCancel"
        android:text="Alert OK Cancel"/>
</LinearLayout>
```

- MainActivity.kt

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

       val txt = findViewById<TextView>(R.id.txtResult)

        findViewById<Button>(R.id.btnOk).setOnClickListener {
            AlertDialog.Builder(this)
                .setMessage("Alert Dialog")
                .setNeutralButton("OK") { _, _ ->
                    txt.text = "Ok button is Clicked"
                }.show()
        }

        findViewById<Button>(R.id.btnYesNo).setOnClickListener {
            AlertDialog.Builder(this)
                .setMessage("Alert Dialog")
                .setPositiveButton("Yes") { _, _ ->
                    txt.text = "Yes button is Clicked"
                }
                .setNegativeButton("No") { _, _ ->
                    txt.text = "No button is Clicked"
                }.show()
        }

        findViewById<Button>(R.id.btnOkCancel).setOnClickListener {
            AlertDialog.Builder(this)
                .setMessage("Alert Dialog")
                .setPositiveButton("OK") { _, _ ->
                    txt.text = "Ok button is Clicked"
                }
                .setNegativeButton("Cancel") { _, _ ->
                    txt.text = "Cancel button is Clicked"
                }.show()
        }
    }
}
```

Program 59,60,61 : This Android application demonstrates data passing between two activities using Intent. The First Activity contains an EditText and a Button.
When the button is clicked, the entered name is sent to the Second Activity.
The Second Activity receives the data and displays it in one or two TextViews.

- Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <EditText
        android:id="@+id/edtName"
        android:hint="Enter Name"/>

    <Button
        android:id="@+id/btnNext"
        android:text="Next"/>
</LinearLayout>
```

- MainActivity.kt

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val edt = findViewById<EditText>(R.id.edtName)

        findViewById<Button>(R.id.btnNext).setOnClickListener {
            val i = Intent(this, SecondActivity::class.java)
            i.putExtra("name", edt.text.toString())
            startActivity(i)
        }
    }
}
```

- Activity_second.xml

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

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
     <TextView
        android:id="@+id/txtHello"
        android:text="Hello"/>

    <TextView
        android:id="@+id/txtName"/>
</LinearLayout>
```

- SecondActivity.kt

```
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import android.widget.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val name = intent.getStringExtra("name")
        findViewById<TextView>(R.id.txtName).text = name
    }
}
```