

Android Practicals QB:

1. Create an Android application to pass the data from one activity to another activity in the same application using intent.

→

Activity_main.xml:

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

    <EditText
        android:id="@+id/editText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Enter some text"
        android:layout_centerInParent="true"/>

    <Button
        android:id="@+id/buttonSendData"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Data"
        android:layout_below="@id/editText"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"/>
</RelativeLayout>
```

Activity_second.xml:

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

    <TextView
        android:id="@+id/textViewReceivedData"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Received data will appear here"
        android:layout_centerInParent="true"/>
</RelativeLayout>
```

MainActivity.kt:

```
package com.example.practical
```

```
import android.content.Intent
import android.os.Bundle
import android.view.View
```

```

import android.widget.Button
import android.widget.EditText
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

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

        val editText = findViewById<EditText>(R.id.editText)
        val buttonSendData = findViewById<Button>(R.id.buttonSendData)

        buttonSendData.setOnClickListener {
            val userInput = editText.text.toString()

            // Create an Intent to start SecondActivity
            val intent = Intent(this, SecondActivity::class.java)

            // Put the data into the Intent
            intent.putExtra("userData", userInput)

            // Start SecondActivity
            startActivity(intent)
        }
    }
}

```

SecondActivity.kt:

```

package com.example.practical

import android.os.Bundle
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class SecondActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_second)

        val textView = findViewById<TextView>(R.id.textViewReceivedData)

        // Get the data from the Intent
        val receivedData = intent.getStringExtra("userData")

        // Set the data to the TextView
        textView.text = receivedData
    }
}

```

AndroidManifest.xml:

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/Theme.AppCompat.DayNight">

        <activity android:name=".SecondActivity" />

        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>

</manifest>
```

2. Write a program using Kotlin to implement control structures and loops. if-else Statement, for Loop, while Loop, do- while Loop.

→

```
fun main() {
    val number = 7
    if (number % 2 == 0) {
        println("$number is even.")
    } else {
        println("$number is odd.")
    }

    println("\nFor Loop Example:")
    for (i in 1..5) {
        println("i = $i")
    }
    println("\nWhile Loop Example:")
    var count = 1
    while (count <= 5) {
        println("count = $count")
        count++
    }

    println("\nDo-While Loop Example:")
```

```

var counter = 1
do {
    println("counter = $counter")
    counter++
} while (counter <= 5)
}

```

3. Create an Android application to display Toast message on pressing the Back button.

→

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"
    android:gravity="center"
    android:padding="16dp">

    <TextView
        android:id="@+id/welcomeText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Press the Back button to see the Toast"
        android:textSize="18sp"
        android:textColor="@android:color/black"
        android:layout_marginBottom="20dp"/>

    <Button
        android:id="@+id/backButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Simulate Back Press"
        android:layout_marginTop="20dp"/>
</LinearLayout>

```

MainActivity.kt:

```

package com.example.practical

import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }
}

```

```

        setContentView(R.layout.activity_main)
    }

    override fun onBackPressed() {
        // Show a Toast message when the back button is pressed
        Toast.makeText(this, "Back button pressed!", Toast.LENGTH_SHORT).show()
    }
}

```

4. Create an Android application using linear layout and insert 10 games in the list view and display the selected game in the text view.

→

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"
    android:padding="16dp">

    <TextView
        android:id="@+id/selectedGameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Selected Game will appear here"
        android:textSize="18sp"
        android:layout_marginBottom="20dp"/>

    <ListView
        android:id="@+id/gamesListView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>

```

MainActivity.kt:

```

package com.example.practical

import android.os.Bundle
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView.OnItemClickListener
import android.widget.ArrayAdapter
import android.widget.ListView
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {

```

```
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)
```

```
val games = listOf(
    "Minecraft", "Fortnite", "Among Us", "Call of Duty", "Apex Legends",
    "Valorant", "FIFA 21", "PUBG Mobile", "GTA V", "Rocket League"
)
```

```
val gamesListView: ListView = findViewById(R.id.gamesListView)
val selectedGameTextView: TextView = findViewById(R.id.selectedGameTextView)
```

```
val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, games)
gamesListView.adapter = adapter
gamesListView.setOnItemClickListener = AdapterView.OnItemClickListener { _, _, position, _ ->
    val selectedGame = games[position]
    selectedGameTextView.text = "Selected Game: $selectedGame"
}
}
```

5. Create an application to create Image Flipper and Image Gallery. On clicking on the image, display the information about the image.

→

**Download any 3 imgs with name image 1, image 2, image 3 and paste them in res/drawable folder.

Activity_main.xml:

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

    <ImageSwitcher
        android:id="@+id/imageSwitcher"
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:src="@drawable/image1"
        android:inAnimation="@android:anim/fade_in"
        android:outAnimation="@android:anim/fade_out" />

    <Button
        android:id="@+id/nextButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Next Image"
        android:layout_below="@id/imageSwitcher"
        android:layout_centerHorizontal="true"
```

```
android:layout_marginTop="20dp" />
```

```
<GridView
    android:id="@+id/gridView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/nextButton"
    android:layout_marginTop="20dp"
    android:columnWidth="100dp"
    android:numColumns="3"
    android:horizontalSpacing="10dp"
    android:verticalSpacing="10dp"
    android:stretchMode="columnWidth"
    android:gravity="center"/>
```

```
</RelativeLayout>
```

ImageAdapter.kt:

```
package com.example.practical
```

```
import android.content.Context
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.BaseAdapter
import android.widget.ImageView
import android.widget.Toast
```

```
class ImageAdapter(private val context: Context, private val imageIds: IntArray) : BaseAdapter() {
```

```
    override fun getCount(): Int {
        return imageIds.size
    }
```

```
    override fun getItem(position: Int): Any {
        return imageIds[position]
    }
```

```
    override fun getItemId(position: Int): Long {
        return position.toLong()
    }
```

```
    override fun getView(position: Int, convertView: View?, parent: ViewGroup?): View {
        val imageView: ImageView
        if (convertView == null) {
            imageView = ImageView(context)
            imageView.layoutParams = ViewGroup.LayoutParams(200, 200)
            imageView.scaleType = ImageView.ScaleType.CENTER_CROP
        } else {
```

```

        imageView = convertView as ImageView
    }

    imageView.setImageResource(imageIds[position])

    imageView.setOnClickListener {
        // Show image info on click
        Toast.makeText(context, "Image $position clicked", Toast.LENGTH_SHORT).show()
    }

    return imageView
}
}

```

MainActivity.kt:

```

package com.example.practical

import android.os.Bundle
import android.widget.ImageSwitcher
import android.widget.Button
import android.widget.GridView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import android.widget.ImageView
import android.widget.AdapterView

class MainActivity : AppCompatActivity() {

    private lateinit var imageSwitcher: ImageSwitcher
    private lateinit var nextButton: Button
    private lateinit var gridView: GridView

    private val imageIds = intArrayOf(
        R.drawable.image1,
        R.drawable.image2,
        R.drawable.image3
    )

    private var currentIndex = 0
    private val imageAdapter: ImageAdapter by lazy {
        ImageAdapter(this, imageIds)
    }

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

        imageSwitcher = findViewById(R.id.imageSwitcher)
        nextButton = findViewById(R.id.nextButton)
    }
}

```



```

gridView = findViewById(R.id.gridView)

// Set up ImageSwitcher for the Image Flipper
imageSwitcher.setFactory {
    ImageView(this).apply {
        setImageResource(imageIds[currentImageIndex])
    }
}

// Button to manually flip the images
nextButton.setOnClickListener {
    currentImageIndex = (currentImageIndex + 1) % imageIds.size
    imageSwitcher.setImageResource(imageIds[currentImageIndex])
}

// Set up the GridView with ImageAdapter for the Image Gallery
gridView.adapter = imageAdapter

// Item click listener to show information when an image is clicked
gridView.setOnItemClickListener { _, _, position, _ ->
    Toast.makeText(this, "Image at position $position clicked", Toast.LENGTH_SHORT).show()
}
}
}

```

6. Write a program using Kotlin to implement control structures and loops. if-else statement, for Loop, while loop, when loop.

→

```

fun main() {
    // If-Else Statement Example
    val number = 10
    if (number > 0) {
        println("The number is positive.")
    } else if (number < 0) {
        println("The number is negative.")
    } else {
        println("The number is zero.")
    }

    // For Loop Example
    println("\nFor Loop Example: Print numbers from 1 to 5")
    for (i in 1..5) {
        println(i)
    }

    // While Loop Example
    println("\nWhile Loop Example: Print numbers less than 5")
    var count = 0
    while (count < 5) {
        println(count)
    }
}

```

```

        count++
    }

    // When Loop Example
    println("\nWhen Loop Example: Check the day of the week")
    val dayOfWeek = 3
    when (dayOfWeek) {
        1 -> println("Monday")
        2 -> println("Tuesday")
        3 -> println("Wednesday")
        4 -> println("Thursday")
        5 -> println("Friday")
        6 -> println("Saturday")
        7 -> println("Sunday")
        else -> println("Invalid day number")
    }
}

```

7. Create an Android application which automatically displays a Toast message when the user turns the Airplane mode on or off using broadcast receiver.

→

activity_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>

```

MyReceiver.kt:

```

package com.example.practical
import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import android.widget.Toast
class MyReceiver : BroadcastReceiver() {

```

```

override fun onReceive(context: Context?, intent: Intent?) {
    if (intent?.action == Intent.ACTION_AIRPLANE_MODE_CHANGED) {
        val isAirplaneModeEnabled = intent.getBooleanExtra("state", false)
        val message = if (isAirplaneModeEnabled) "Airplane Mode Enabled" else "Airplane Mode Disabled"
        Toast.makeText(context, message, Toast.LENGTH_LONG).show()
    }
}
}
}

```

AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.Practical"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".MyReceiver"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.AIRPLANE_MODE" />
            </intent-filter>
        </receiver>
    </application>
</manifest>

```

MainActivity.kt:

```

package com.example.practical
import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat

```

```

import androidx.core.view.WindowInsetsCompat
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
        }
    }
}

```

8. Write an Android application demonstrating response to event/user interaction for Checkbox, Radio button, Button, Spinner.

→

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"
    android:padding="16dp"
    android:gravity="center">

    <CheckBox
        android:id="@+id/checkbox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I agree to the terms and conditions" />

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <RadioButton
            android:id="@+id/radioOption1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 1" />

        <RadioButton
            android:id="@+id/radioOption2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 2" />
    </RadioGroup>
</LinearLayout>

```

```
<RadioButton
    android:id="@+id/radioOption3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Option 3" />
</RadioGroup>
```

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click Me" />
```

```
<Spinner
    android:id="@+id/spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
</LinearLayout>
```

MainActivity.kt:

```
package com.example.practical
```

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

```
class MainActivity : AppCompatActivity() {

    private lateinit var checkbox: CheckBox
    private lateinit var radioGroup: RadioGroup
    private lateinit var radioButton1: RadioButton
    private lateinit var radioButton2: RadioButton
    private lateinit var radioButton3: RadioButton
    private lateinit var button: Button
    private lateinit var spinner: Spinner

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

        checkbox = findViewById(R.id.checkbox)
        radioGroup = findViewById(R.id.radioGroup)
        radioButton1 = findViewById(R.id.radioOption1)
        radioButton2 = findViewById(R.id.radioOption2)
        radioButton3 = findViewById(R.id.radioOption3)
        button = findViewById(R.id.button)
```

```
spinner = findViewById(R.id.spinner)
```

```
val spinnerItems = arrayOf("Option A", "Option B", "Option C")
```

```
val spinnerAdapter = ArrayAdapter(this, android.R.layout.simple_spinner_item, spinnerItems)
```

```
spinnerAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item)
```

```
spinner.adapter = spinnerAdapter
```

```
checkbox.setOnCheckedChangeListener { _, isChecked ->
```

```
    if (isChecked) {
```

```
        Toast.makeText(this, "Checkbox Checked", Toast.LENGTH_SHORT).show()
```

```
    } else {
```

```
        Toast.makeText(this, "Checkbox Unchecked", Toast.LENGTH_SHORT).show()
```

```
    }
```

```
}
```

```
radioGroup.setOnCheckedChangeListener { _, checkedId ->
```

```
    when (checkedId) {
```

```
        R.id.radioOption1 -> Toast.makeText(this, "Option 1 Selected", Toast.LENGTH_SHORT).show()
```

```
        R.id.radioOption2 -> Toast.makeText(this, "Option 2 Selected", Toast.LENGTH_SHORT).show()
```

```
        R.id.radioOption3 -> Toast.makeText(this, "Option 3 Selected", Toast.LENGTH_SHORT).show()
```

```
    }
```

```
}
```

```
button.setOnClickListener {
```

```
    Toast.makeText(this, "Button Clicked", Toast.LENGTH_SHORT).show()
```

```
}
```

```
spinner.onItemSelectedListener = object : AdapterView.OnItemSelectedListener {
```

```
    override fun onItemSelected(parentView: AdapterView<*>?, view: View?, position: Int, id: Long) {
```

```
        val selectedItem = parentView?.getItemAtPosition(position) as String
```

```
        Toast.makeText(this@MainActivity, "$selectedItem Selected", Toast.LENGTH_SHORT).show()
```

```
    }
```

```
    override fun onNothingSelected(parentView: AdapterView<*>?) {
```

```
    }
```

```
}
```

```
}
```

```
}
```

9. Create the media API in Android to play an audio file.

→

**Add an Audio file named audio_sample.mp3 to res/raw

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"
android:gravity="center"
android:padding="16dp">
```

```
<Button
    android:id="@+id/playButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Play Audio" />
```

```
<SeekBar
    android:id="@+id/seekBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="100"
    android:progress="0"
    android:enabled="false" />
```

```
</LinearLayout>
```

MainActivity.kt:

```
package com.example.practical
```

```
import android.media.MediaPlayer
import android.os.Bundle
import android.os.Handler
import android.widget.Button
import android.widget.SeekBar
import androidx.appcompat.app.AppCompatActivity
```

```
class MainActivity : AppCompatActivity() {
```

```
    private lateinit var mediaPlayer: MediaPlayer
    private lateinit var playButton: Button
    private lateinit var seekBar: SeekBar
    private val handler = Handler()
```

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

```
        playButton = findViewById(R.id.playButton)
        seekBar = findViewById(R.id.seekBar)
```

```
        mediaPlayer = MediaPlayer.create(this, R.raw.audio_sample)
```

```
        playButton.setOnClickListener {
            if (mediaPlayer.isPlaying) {
                mediaPlayer.pause()
            }
        }
```

```

        playButton.text = "Play Audio"
    } else {
        mediaPlayer.start()
        playButton.text = "Pause Audio"
        updateSeekBar()
    }
}

seekBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {
    override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {
        if (fromUser) {
            mediaPlayer.seekTo(progress)
        }
    }

    override fun onStartTrackingTouch(seekBar: SeekBar?) {}
    override fun onStopTrackingTouch(seekBar: SeekBar?) {}
}))

// Listen to when the media file is completed
mediaPlayer.setOnCompletionListener {
    playButton.text = "Play Audio"
    seekBar.progress = 0
}
}

private fun updateSeekBar() {
    seekBar.max = mediaPlayer.duration
    handler.postDelayed(object : Runnable {
        override fun run() {
            seekBar.progress = mediaPlayer.currentPosition
            handler.postDelayed(this, 1000)
        }
    }, 0)
}

override fun onPause() {
    super.onPause()
    mediaPlayer.release()
}
}

```

10. Create an Android application, with one button and an image view, also set the background image in the application, the image should be changed on each click of the button (Use two images).

→

** Add an images named image 1 and image 2 to res/drawable folder.

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"
android:gravity="center"
android:padding="16dp"
android:background="@drawable/image1">
```

```
<Button
    android:id="@+id/changeImageButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Change Background Image"
    android:textSize="18sp"/>
```

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:src="@drawable/image1" />
```

```
</LinearLayout>
```

MainActivity.kt:

```
package com.example.practical
```

```
import android.os.Bundle
import android.widget.Button
import android.widget.ImageView
import androidx.appcompat.app.AppCompatActivity
```

```
class MainActivity : AppCompatActivity() {
```

```
    private lateinit var changeImageButton: Button
    private lateinit var imageView: ImageView
    private var isImage1 = true
```

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

```
        changeImageButton = findViewById(R.id.changeImageButton)
        imageView = findViewById(R.id.imageView)
```

```
        changeImageButton.setOnClickListener {
            if (isImage1) {
                findViewById<LinearLayout>(R.id.layout).setBackgroundResource(R.drawable.image2)
                imageView.setImageResource(R.drawable.image2)
            } else {
```

```

        findViewById<LinearLayout>(R.id.layout).setBackgroundResource(R.drawable.image1)
        imageView.setImageResource(R.drawable.image1)
    }
    isImage1 = !isImage1
}
}
}
}

```

11. Create an Android application using linear layout and insert 10 animals in the list view and display the appropriate Toast.

→

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"
    android:padding="16dp"
    android:gravity="center">

    <ListView
        android:id="@+id/animalListView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>

```

MainActivity.kt:

```

package com.example.practical

import android.os.Bundle
import android.view.View
import android.widget.AdapterView
import android.widget.AdapterView.OnItemClickListener
import android.widget.ArrayAdapter
import android.widget.ListView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

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

        val animalListView: ListView = findViewById(R.id.animalListView)

        val animals = arrayOf(
            "Lion", "Tiger", "Elephant", "Zebra", "Giraffe",
            "Monkey", "Bear", "Kangaroo", "Panda", "Koala"

```

)

```
val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, animals)
animalListView.adapter = adapter
```

```
animalListView.setOnItemClickListener { parent, view, position, id ->
    val selectedAnimal = parent.getItemAtPosition(position).toString()
    Toast.makeText(this, "Selected: $selectedAnimal", Toast.LENGTH_SHORT).show()
}
}
```

12. Create an Android application to demonstrate the use of a sub menu. The toast should appear by selecting the sub menu item.

→

****Create menu file in res/menu folder.**

menu_main.xml:

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

    <item
        android:id="@+id/menu_cut"
        android:title="Cut"
        android:icon="@android:drawable/ic_menu_cut"
        android:showAsAction="ifRoom"/>

    <item
        android:id="@+id/menu_copy"
        android:title="Copy"
        android:icon="@android:drawable/ic_menu_copy"
        android:showAsAction="ifRoom"/>

    <item
        android:id="@+id/menu_paste"
        android:title="Paste"
        android:icon="@android:drawable/ic_menu_paste"
        android:showAsAction="ifRoom"/>

</menu>
```

MainActivity.kt:

```
package com.example.practical
```

```
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
```

```

class MainActivity : AppCompatActivity() {

    private lateinit var editText: EditText

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

        editText = findViewById(R.id.editText)
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu_main, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        when (item.itemId) {
            R.id.menu_cut -> {
                val cutText = editText.text.toString()
                editText.setText("") // Clear the text after cut
                Toast.makeText(this, "Cut: $cutText", Toast.LENGTH_SHORT).show()
                return true
            }
            R.id.menu_copy -> {
                val copyText = editText.text.toString()
                Toast.makeText(this, "Copied: $copyText", Toast.LENGTH_SHORT).show()
                return true
            }
            R.id.menu_paste -> {
                val pasteText = "Pasted Text"
                editText.append(pasteText)
                Toast.makeText(this, "Pasted: $pasteText", Toast.LENGTH_SHORT).show()
                return true
            }
            else -> return super.onOptionsItemSelected(item)
        }
    }
}

```

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"
    android:padding="16dp"

```

```
android:gravity="center">
```

```
<EditText
```

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

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

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

```
    android:hint="Type here"
```

```
    android:padding="16dp"
```

```
    android:textSize="18sp"
```

```
    android:layout_marginBottom="20dp"/>
```

```
</LinearLayout>
```