1) Background music (Service + MediaPlayer)

MusicService.java

```
package com.example.madpracticals;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.IBinder;
import androidx.annotation.Nullable;

public class MusicService extends Service {
    private MediaPlayer player;

    @Override
    public void onCreate() {
        super.onCreate();
        player = MediaPlayer.create(this, R.raw.bg_music); // place
bg_music.mp3 in res/raw
        player.setLooping(true);
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        if (player != null && !player.isPlaying()) player.start();
        return START_STICKY;
    }

    @Override
    public void onDestroy() {
        if (player != null) {
            player.stop();
            player.release();
            player = null;
        }
        super.onDestroy();
    }

    @Nullable
    @Override
```

```
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```

MainActivity.java

```
package com.example.madpracticals;

import android.content.Intent;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityMainBinding;

public class MainActivity extends AppCompatActivity {
    private ActivityMainBinding b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnStart.setOnClickListener(v -> startService(new
Intent(this, MusicService.class)));
        b.btnStop.setOnClickListener(v -> stopService(new Intent(this,
MusicService.class)));
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="24dp"
    android:layout_width="match_parent"
android:layout_height="match_parent">

    <Button android:id="@+id/btnStart"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Start Music"/>
    <Button android:id="@+id/btnStop"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Stop Music"
```

```
android:layout_marginTop="12dp"/>
</LinearLayout>
```

Manifest note: Add <service android:name=".MusicService" /> inside <application>. Place bg_music.mp3 under res/raw.

2) Student SQLite app — Insert 5 records & show all

StudentDbHelper.java

```
package com.example.madpracticals;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

public class StudentDbHelper extends SQLiteOpenHelper {
    private static final String DB_NAME = "students.db";
    private static final int DB_VER = 1;
    public static final String TABLE = "Student";
    public StudentDbHelper(@Nullable Context context) { super(context,
DB_NAME, null, DB_VER); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String sql = "CREATE TABLE " + TABLE + " (roll INTEGER PRIMARY
KEY, name TEXT, address TEXT, percentage REAL)";
        db.execSQL(sql);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldV, int newV) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE);
        onCreate(db);
    }

    public void insertStudent(int roll, String name, String address,
double percent) {
        SQLiteDatabase db = getWritableDatabase();
        ContentValues cv = new ContentValues();
        cv.put("roll", roll); cv.put("name", name); cv.put("address",
```

```
address); cv.put("percentage", percent);
        db.insert(TABLE, null, cv);
    }

    public Cursor getAllStudents() {
        SQLiteDatabase db = getReadableDatabase();
        return db.rawQuery("SELECT * FROM " + TABLE + " ORDER BY roll",
null);
    }
}
```

StudentListActivity.java

```
package com.example.madpracticals;

import android.database.Cursor;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import androidx.appcompat.app.AppCompatActivity;
import
com.example.madpracticals.databinding.ActivityStudentListBinding;
import java.util.ArrayList;

public class StudentListActivity extends AppCompatActivity {
    private ActivityStudentListBinding b;
    private StudentDbHelper db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityStudentListBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());
        db = new StudentDbHelper(this);

        // Insert 5 students if DB empty
        Cursor c = db.getAllStudents();
        if (!c.moveToFirst()) {
            db.insertStudent(1, "Aman", "Pune", 72.5);
            db.insertStudent(2, "Riya", "Mumbai", 81.0);
            db.insertStudent(3, "Rahul", "Nashik", 65.2);
            db.insertStudent(4, "Priya", "Kolhapur", 78.3);
            db.insertStudent(5, "Sahil", "Pimpri", 90.0);
        }
        c.close();
```

```
        loadList();
        b.btnRefresh.setOnClickListener(v -> loadList());
    }

    private void loadList() {
        Cursor c = db.getAllStudents();
        ArrayList<String> arr = new ArrayList<>();
        while (c.moveToNext()) {
            int roll = c.getInt(c.getColumnIndexOrThrow("roll"));
            String name = c.getString(c.getColumnIndexOrThrow("name"));
            String addr =
c.getString(c.getColumnIndexOrThrow("address"));
            double p =
c.getDouble(c.getColumnIndexOrThrow("percentage"));
            arr.add(roll + " - " + name + " - " + addr + " - " + p +
"%");
        }
        c.close();
        b.listView.setAdapter(new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, arr));
    }
}
```

activity_student_list.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="16dp"
    android:layout_width="match_parent"
android:layout_height="match_parent">

    <Button android:id="@+id/btnRefresh"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Refresh"/>
    <ListView android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="match_parent" android:layout_marginTop="8dp"/>
</LinearLayout>
```

Manifest note: No special permissions required.

3) Prime-number checker

PrimeActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityPrimeBinding;

public class PrimeActivity extends AppCompatActivity {
    private ActivityPrimeBinding b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityPrimeBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnCheck.setOnClickListener(v -> {
            String s = b.etNumber.getText().toString().trim();
            if (s.isEmpty()) { Toast.makeText(this, "Enter number",
Toast.LENGTH_SHORT).show(); return; }
            try {
                long n = Long.parseLong(s);
                boolean prime = isPrime(n);
                b.tvResult.setText(n + (prime ? " is Prime." : " is NOT
Prime."));
            } catch (NumberFormatException e) {
                Toast.makeText(this, "Invalid number",
Toast.LENGTH_SHORT).show();
            }
        });
    }

    private boolean isPrime(long n) {
        if (n < 2) return false;
        if (n % 2 == 0) return n == 2;
        for (long i = 3; i * i <= n; i += 2) if (n % i == 0) return
false;
        return true;
    }
}
```

activity_prime.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="24dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <EditText android:id="@+id/etNumber" android:inputType="number"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:hint="Enter number"/>
    <Button android:id="@+id/btnCheck"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Check Prime"
android:layout_marginTop="12dp"/>
    <TextView android:id="@+id/tvResult"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:textSize="18sp"
android:layout_marginTop="12dp"/>
</LinearLayout>
```

Manifest note: No permissions required.

4) Simple Calculator (add, sub, mul, div)

CalculatorActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityCalculatorBinding;

public class CalculatorActivity extends AppCompatActivity {
    private ActivityCalculatorBinding b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityCalculatorBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnAdd.setOnClickListener(v -> calculate('+'));
        b.btnSub.setOnClickListener(v -> calculate('-'));
```

```
        b.btnMul.setOnClickListener(v -> calculate('*'));
        b.btnDiv.setOnClickListener(v -> calculate('/'));
    }

    private void calculate(char op) {
        try {
            double a =
Double.parseDouble(b.etA.getText().toString().trim());
            double c =
Double.parseDouble(b.etB.getText().toString().trim());
            double res;
            switch (op) {
                case '+': res = a + c; break;
                case '-': res = a - c; break;
                case '*': res = a * c; break;
                case '/':
                    if (c == 0) { Toast.makeText(this, "Cannot divide
by zero", Toast.LENGTH_SHORT).show(); return; }
                    res = a / c; break;
                default: return;
            }
            b.tvResult.setText("Result: " + res);
        } catch (NumberFormatException e) {
            Toast.makeText(this, "Enter valid numbers",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

activity_calculator.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="16dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <EditText android:id="@+id/etA" android:layout_width="match_parent"
android:layout_height="wrap_content" android:inputType="numberDecimal"
android:hint="Number A"/>
    <EditText android:id="@+id/etB" android:layout_width="match_parent"
android:layout_height="wrap_content" android:inputType="numberDecimal"
android:hint="Number B" android:layout_marginTop="8dp"/>
    <LinearLayout android:orientation="horizontal"
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content" android:layout_marginTop="12dp">
        <Button android:id="@+id/btnAdd" android:layout_width="0dp"
android:layout_weight="1" android:layout_height="wrap_content"
android:text="+"/>
        <Button android:id="@+id/btnSub" android:layout_width="0dp"
android:layout_weight="1" android:layout_height="wrap_content"
android:text="-"/>
        <Button android:id="@+id/btnMul" android:layout_width="0dp"
android:layout_weight="1" android:layout_height="wrap_content"
android:text="*"/>
        <Button android:id="@+id/btnDiv" android:layout_width="0dp"
android:layout_weight="1" android:layout_height="wrap_content"
android:text="/"/>
    </LinearLayout>
    <TextView android:id="@+id/tvResult"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:textSize="18sp"
android:layout_marginTop="16dp"/>
</LinearLayout>
```

Manifest note: No permissions required.

5) GridView demo

GridActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityGridBinding;

public class GridActivity extends AppCompatActivity {
    private ActivityGridBinding b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityGridBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        String[] fruits = {"Apple", "Banana", "Cherry", "Date",
"Grapes", "Mango", "Orange", "Papaya"};
```

```
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, fruits);
        b.gridView.setAdapter(adapter);
        b.gridView.setNumColumns(2);
    }
}
```

activity_grid.xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/gridView" android:layout_width="match_parent"
android:layout_height="match_parent"
    android:columnWidth="120dp" android:verticalSpacing="10dp"
android:horizontalSpacing="10dp" android:gravity="center"/>
```

Manifest note: No permissions required.

6) Employee SQLite (insert 5 + show)

EmployeeDbHelper.java

```
package com.example.madpracticals;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

public class EmployeeDbHelper extends SQLiteOpenHelper {
    private static final String DB_NAME = "employees.db";
    private static final int DB_VER = 1;
    public static final String TABLE = "Employee";

    public EmployeeDbHelper(@Nullable Context context) { super(context,
DB_NAME, null, DB_VER); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE " + TABLE + " (E_id INTEGER PRIMARY
KEY, name TEXT, address TEXT, pho_no TEXT)");
    }

    @Override
```

```
    public void onUpgrade(SQLiteDatabase db, int oldV, int newV) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE);
        onCreate(db);
    }

    public void insertEmployee(int id, String name, String addr, String
phone) {
        ContentValues cv = new ContentValues();
        cv.put("E_id", id); cv.put("name", name); cv.put("address",
addr); cv.put("pho_no", phone);
        getWritableDatabase().insert(TABLE, null, cv);
    }

    public Cursor getAll() { return
getReadableDatabase().rawQuery("SELECT * FROM " + TABLE + " ORDER BY
E_id", null); }
}
```

EmployeeActivity.java

```
package com.example.madpracticals;

import android.database.Cursor;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityEmployeeBinding;
import java.util.ArrayList;

public class EmployeeActivity extends AppCompatActivity {
    private ActivityEmployeeBinding b;
    private EmployeeDbHelper db;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityEmployeeBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());
        db = new EmployeeDbHelper(this);

        Cursor c = db.getAll();
        if (!c.moveToFirst()) {
            db.insertEmployee(1,"Amit","Pune","9876543210");
            db.insertEmployee(2,"Sneha","Mumbai","9876501234");
            db.insertEmployee(3,"Vikram","Aurangabad","9123456780");
```

```
            db.insertEmployee(4,"Kavya","Nagpur","9001234567");
            db.insertEmployee(5,"Deep","Solapur","7890123456");
        }
        c.close();
        load();
        b.btnRefresh.setOnClickListener(v -> load());
    }

    private void load() {
        Cursor c = db.getAll();
        ArrayList<String> arr = new ArrayList<>();
        while (c.moveToNext()) {
            arr.add(c.getInt(c.getColumnIndexOrThrow("E_id")) + " - " +
                    c.getString(c.getColumnIndexOrThrow("name")) + " -
" +
                    c.getString(c.getColumnIndexOrThrow("address")) + "
- " +
                    c.getString(c.getColumnIndexOrThrow("pho_no")));
        }
        c.close();
        b.listView.setAdapter(new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, arr));
    }
}
```

activity_employee.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="16dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <Button android:id="@+id/btnRefresh"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Refresh"/>
    <ListView android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="match_parent" android:layout_marginTop="8dp"/>
</LinearLayout>
```

Manifest note: No permissions required.

7) Activity lifecycle demo

LifecycleActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.util.Log;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class LifecycleActivity extends AppCompatActivity {
    private static final String TAG = "LifecycleDemo";
    private TextView tv;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        tv = new TextView(this);
        tv.setPadding(24,24,24,24);
        tv.setText("Lifecycle log will appear in Logcat and UI");
        setContentView(tv);
        log("onCreate");
    }

    @Override protected void onStart(){ super.onStart();
log("onStart"); }
    @Override protected void onResume(){ super.onResume();
log("onResume"); }
    @Override protected void onPause(){ super.onPause();
log("onPause"); }
    @Override protected void onStop(){ super.onStop(); log("onStop"); }
    @Override protected void onRestart(){ super.onRestart();
log("onRestart"); }
    @Override protected void onDestroy(){ super.onDestroy();
log("onDestroy"); }

    private void log(String s) {
        Log.d(TAG, s);
        tv.append("
" + s);
    }
}
```

Manifest note: No separate layout file; activity creates UI programmatically.

8) Power & Average: two Activities

ComputeActivity.java

```
package com.example.madpracticals;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityComputeBinding;

public class ComputeActivity extends AppCompatActivity {
    private ActivityComputeBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityComputeBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());
        b.btnPower.setOnClickListener(v -> doOp("power"));
        b.btnAverage.setOnClickListener(v -> doOp("avg"));
    }
    private void doOp(String op) {
        try {
            double a =
Double.parseDouble(b.etA.getText().toString().trim());
            double bval =
Double.parseDouble(b.etB.getText().toString().trim());
            Intent i = new Intent(this, ResultActivity.class);
            if (op.equals("power")) {
                i.putExtra("result", Math.pow(a, (int)bval));
                i.putExtra("type", "Power");
            } else {
                i.putExtra("result", (a + bval) / 2.0);
                i.putExtra("type", "Average");
            }
            startActivity(i);
        } catch (NumberFormatException e) {
            Toast.makeText(this, "Enter valid numbers",
Toast.LENGTH_SHORT).show();
        }
```

```
    }
}
```

ResultActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityResultBinding;

public class ResultActivity extends AppCompatActivity {
    private ActivityResultBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityResultBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());
        double res = getIntent().getDoubleExtra("result", 0.0);
        String type = getIntent().getStringExtra("type");
        b.tvResult.setText(type + ": " + res);
    }
}
```

activity_compute.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="18dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <EditText android:id="@+id/etA" android:inputType="numberDecimal"
android:hint="Number A" android:layout_width="match_parent"
android:layout_height="wrap_content"/>
    <EditText android:id="@+id/etB" android:inputType="numberDecimal"
android:hint="Number B" android:layout_width="match_parent"
android:layout_height="wrap_content" android:layout_marginTop="8dp"/>
    <Button android:id="@+id/btnPower" android:text="Power"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:layout_marginTop="10dp"/>
    <Button android:id="@+id/btnAverage" android:text="Average"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:layout_marginTop="8dp"/>
</LinearLayout>
```

activity_result.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="24dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <TextView android:id="@+id/tvResult"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:textSize="20sp"/>
</LinearLayout>
```

Manifest note: Add ComputeActivity and ResultActivity to AndroidManifest.xml.

9) JSON Employee info (create local JSON, parse, display in ListView)

JsonActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityJsonBinding;
import org.json.JSONArray;
import org.json.JSONObject;
import java.util.ArrayList;

public class JsonActivity extends AppCompatActivity {
    private ActivityJsonBinding b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityJsonBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        String json = "{ "employees": [ " +
                "{"id":1,"name":"Aman","desg":"Manager"}," +
                "{"id":2,"name":"Riya","desg":"Developer"}," +
                "{"id":3,"name":"Rahul","desg":"Tester"}" +
                "]}";

        ArrayList<String> arr = new ArrayList<>();
```

```
        try {
            JSONObject root = new JSONObject(json);
            JSONArray emps = root.getJSONArray("employees");
            for (int i=0;i<emps.length();i++){
                JSONObject e = emps.getJSONObject(i);
                arr.add(e.getInt("id") + " - " + e.getString("name") +
" (" + e.getString("desg") + ")");
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        b.listView.setAdapter(new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, arr));
    }
}
```

activity_json.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="16dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <ListView android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="match_parent"/>
</LinearLayout>
```

Manifest note: No permissions required.

10) Armstrong & Perfect number checker

NumberCheckActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import
com.example.madpracticals.databinding.ActivityNumberCheckBinding;

public class NumberCheckActivity extends AppCompatActivity {
    private ActivityNumberCheckBinding b;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityNumberCheckBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnCheckArmstrong.setOnClickListener(v -> {
            String s = b.etNumber.getText().toString().trim();
            if (s.isEmpty()) { Toast.makeText(this,"Enter
number",Toast.LENGTH_SHORT).show(); return; }
            int n = Integer.parseInt(s);
            b.tvArmResult.setText(n + (isArmstrong(n) ? " is Armstrong"
: " is NOT Armstrong"));
        });

        b.btnCheckPerfect.setOnClickListener(v -> {
            String s = b.etNumber.getText().toString().trim();
            if (s.isEmpty()) { Toast.makeText(this,"Enter
number",Toast.LENGTH_SHORT).show(); return; }
            int n = Integer.parseInt(s);
            b.tvPerfectResult.setText(n + (isPerfect(n) ? " is Perfect"
: " is NOT Perfect"));
        });
    }

    private boolean isArmstrong(int n) {
        int temp = n, sum = 0;
        int digits = String.valueOf(n).length();
        while (temp > 0) {
            int d = temp % 10;
            sum += Math.pow(d, digits);
            temp /= 10;
        }
        return sum == n;
    }

    private boolean isPerfect(int n) {
        if (n <= 1) return false;
        int sum = 1;
        for (int i = 2; i * i <= n; i++) {
            if (n % i == 0) {
                sum += i + (i * i == n ? 0 : n / i);
            }
```

```
        }
        return sum == n;
    }
}
```

activity_number_check.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="18dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <EditText android:id="@+id/etNumber" android:inputType="number"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:hint="Enter number"/>
    <Button android:id="@+id/btnCheckArmstrong"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Check Armstrong"
android:layout_marginTop="8dp"/>
    <TextView android:id="@+id/tvArmResult"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:layout_marginTop="6dp"/>
    <Button android:id="@+id/btnCheckPerfect"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Check Perfect"
android:layout_marginTop="12dp"/>
    <TextView android:id="@+id/tvPerfectResult"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:layout_marginTop="6dp"/>
</LinearLayout>
```

Manifest note: No permissions required.