write an android program for Internal storage and external storage (Input the file name (edit text) and contents (multi line edit text), 4 buttons), read and show the contents in the next activity text view:

Code:

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">

    <EditText
        android:id="@+id/editFileName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter file name" />

    <EditText
        android:id="@+id/editContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter content"
        android:inputType="textMultiLine"
        android:minLines="5"
        android:gravity="top"/>

    <Button
        android:id="@+id/btnSaveInternal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save to Internal Storage" />

    <Button
        android:id="@+id/btnLoadInternal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Load from Internal Storage" />

    <Button
        android:id="@+id/btnSaveExternal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save to External Storage" />
    <Button
        android:id="@+id/btnLoadExternal"
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:text="Load from External Storage" />
</LinearLayout>
```

MainActivity.java:

```
package com.example.intext;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    EditText editFileName, editContent;
    Button btnSaveInternal, btnLoadInternal, btnSaveExternal, btnLoadExternal;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editFileName = findViewById(R.id.editFileName);
        editContent = findViewById(R.id.editContent);
        btnSaveInternal = findViewById(R.id.btnSaveInternal);
        btnLoadInternal = findViewById(R.id.btnLoadInternal);
        btnSaveExternal = findViewById(R.id.btnSaveExternal);
        btnLoadExternal = findViewById(R.id.btnLoadExternal);

        btnSaveInternal.setOnClickListener(v -> saveToInternal());
        btnLoadInternal.setOnClickListener(v -> loadFromInternal());
        btnSaveExternal.setOnClickListener(v -> saveToExternal());
        btnLoadExternal.setOnClickListener(v -> loadFromExternal());
    }

    private void saveToInternal() {
        String fileName = editFileName.getText().toString();
        String content = editContent.getText().toString();

        try (FileOutputStream fos = openFileOutput(fileName, MODE_PRIVATE)) {
            fos.write(content.getBytes());
            Toast.makeText(this, "Saved to internal storage",
```

```
Toast.LENGTH_SHORT).show();
        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(this, "Failed to save to internal storage",
Toast.LENGTH_SHORT).show();
        }
    }

    private void loadFromInternal() {
        String fileName = editFileName.getText().toString();

        try (FileInputStream fis = openFileInput(fileName)) {
            byte[] data = new byte[fis.available()];
            fis.read(data);
            String content = new String(data);

            Intent intent = new Intent(this, DisplayActivity.class);
            intent.putExtra("content", content);
            startActivity(intent);
        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(this, "Failed to load from internal storage",
Toast.LENGTH_SHORT).show();
        }
    }

    private void saveToExternal() {
        if (!isExternalStorageWritable()) {
            Toast.makeText(this, "External storage not available",
Toast.LENGTH_SHORT).show();
            return;
        }

        String fileName = editFileName.getText().toString();
        String content = editContent.getText().toString();

        File file = new File(getExternalFilesDir(null), fileName);

        try (FileOutputStream fos = new FileOutputStream(file)) {
            fos.write(content.getBytes());
            Toast.makeText(this, "Saved to external storage",
Toast.LENGTH_SHORT).show();
        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(this, "Failed to save to external storage",
Toast.LENGTH_SHORT).show();
        }
    }

    private void loadFromExternal() {
```

```
        if (!isExternalStorageReadable()) {
            Toast.makeText(this, "External storage not available",
Toast.LENGTH_SHORT).show();
            return;
        }

        String fileName = editFileName.getText().toString();
        File file = new File(getExternalFilesDir(null), fileName);

        try (FileInputStream fis = new FileInputStream(file)) {
            byte[] data = new byte[fis.available()];
            fis.read(data);
            String content = new String(data);

            Intent intent = new Intent(this, DisplayActivity.class);
            intent.putExtra("content", content);
            startActivity(intent);
        } catch (IOException e) {
            e.printStackTrace();
            Toast.makeText(this, "Failed to load from external storage",
Toast.LENGTH_SHORT).show();
        }
    }

    private boolean isExternalStorageWritable() {
        return
Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED);
    }

    private boolean isExternalStorageReadable() {
        String state = Environment.getExternalStorageState();
        return state.equals(Environment.MEDIA_MOUNTED) ||
state.equals(Environment.MEDIA_MOUNTED_READ_ONLY);
    }
}
```

DisplayActivity.java:

```
package com.example.intext;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class DisplayActivity extends AppCompatActivity {

    TextView textViewContent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display);

        textViewContent = findViewById(R.id.textViewContent);

        // Get the content passed from MainActivity
        String content = getIntent().getStringExtra("content");
        textViewContent.setText(content);
    }
}
```

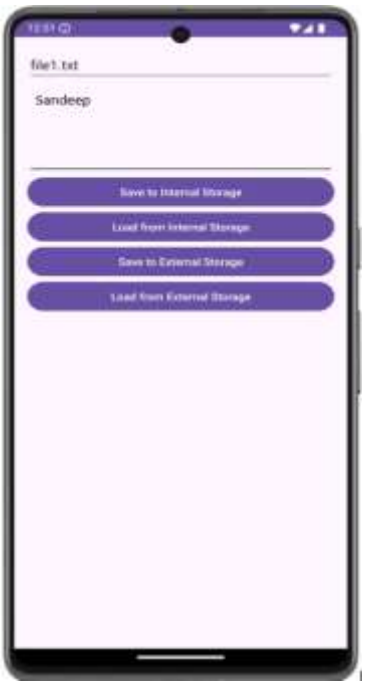
activity_display.xml:

```
<?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/textViewContent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="File content will be displayed here"
        android:textSize="16sp" />

</LinearLayout>
```

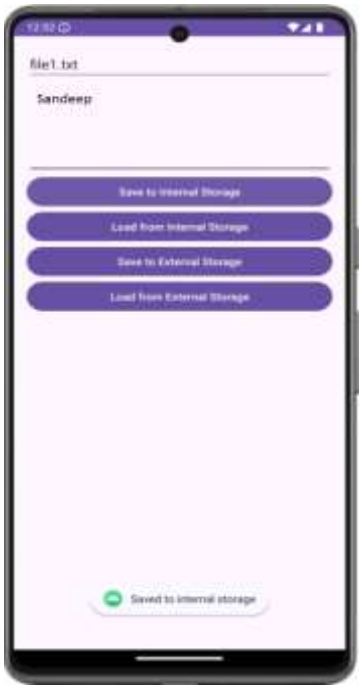
AndroidManifest.xml:

```
<?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.WRITE_EXTERNAL_STORAGE"
        android:maxSdkVersion="28" />
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <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:supportsRtl="true"
        android:theme="@style/Theme.Intext"
        tools:targetApi="31">
        <activity
            android:name=".DisplayActivity"
            android:exported="false" />
        <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>
```

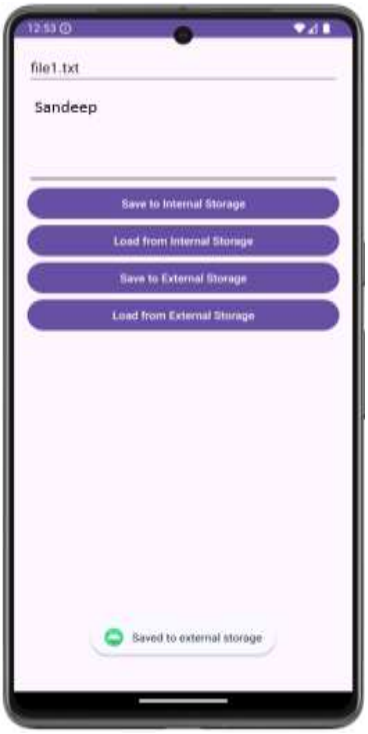
OUTPUT:



Saved to internal:



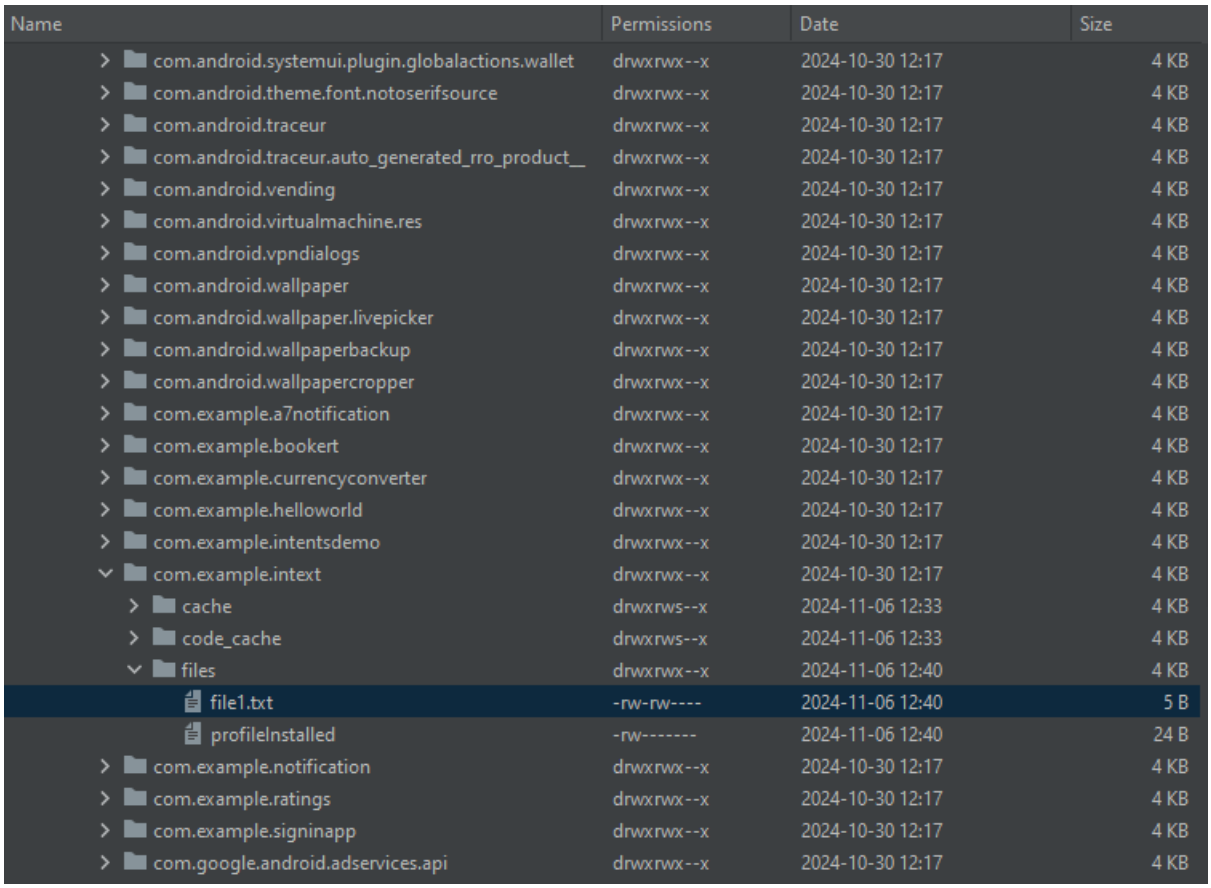
Saved to external:



Reading file1.txt from internal storage:

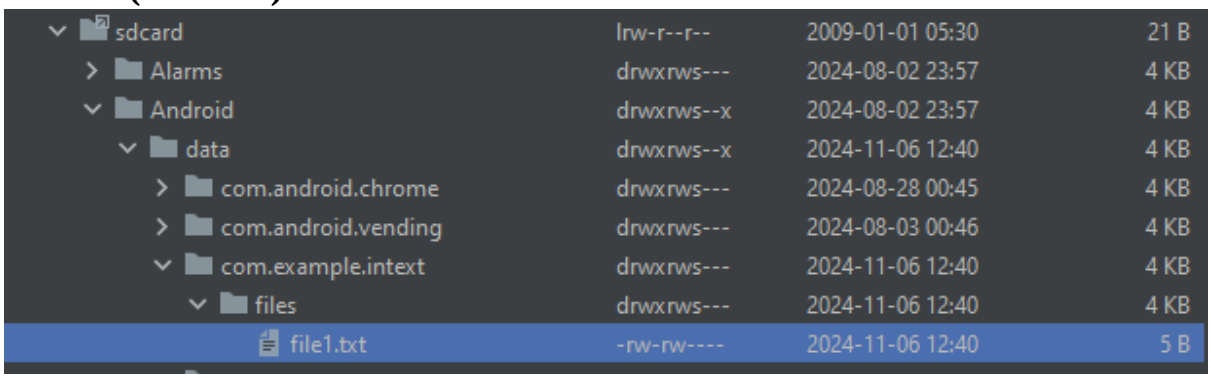
Device explorer:

Internal:



Name	Permissions	Date	Size
com.android.systemui.plugin.globalactions.wallet	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.theme.font.notoserifsource	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.traceur	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.traceur.auto_generated_rro_product__	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.vending	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.virtualmachine.res	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.vpndialogs	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.wallpaper	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.wallpaper.livepicker	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.wallpaperbackup	drwxrwx--x	2024-10-30 12:17	4 KB
com.android.wallpapercropper	drwxrwx--x	2024-10-30 12:17	4 KB
com.example.a7notification	drwxrwx--x	2024-10-30 12:17	4 KB
com.example.bookert	drwxrwx--x	2024-10-30 12:17	4 KB
com.example.currencyconverter	drwxrwx--x	2024-10-30 12:17	4 KB
com.example.helloworld	drwxrwx--x	2024-10-30 12:17	4 KB
com.example.intentsdemo	drwxrwx--x	2024-10-30 12:17	4 KB
com.example.intext	drwxrwx--x	2024-10-30 12:17	4 KB
cache	drwxrws--x	2024-11-06 12:33	4 KB
code_cache	drwxrws--x	2024-11-06 12:33	4 KB
files	drwxrwx--x	2024-11-06 12:40	4 KB
file1.txt	-rw-rw----	2024-11-06 12:40	5 B
profileInstalled	-rw-------	2024-11-06 12:40	24 B
com.example.notification	drwxrwx--x	2024-10-30 12:17	4 KB
com.example.ratings	drwxrwx--x	2024-10-30 12:17	4 KB
com.example.signinapp	drwxrwx--x	2024-10-30 12:17	4 KB
com.google.android.adservices.api	drwxrwx--x	2024-10-30 12:17	4 KB

Sdcard (external):



sdcard	lrw-r--r--	2009-01-01 05:30	21 B
Alarms	drwxrws---	2024-08-02 23:57	4 KB
Android	drwxrws--x	2024-08-02 23:57	4 KB
data	drwxrws--x	2024-11-06 12:40	4 KB
com.android.chrome	drwxrws---	2024-08-28 00:45	4 KB
com.android.vending	drwxrws---	2024-08-03 00:46	4 KB
com.example.intext	drwxrws---	2024-11-06 12:40	4 KB
files	drwxrws---	2024-11-06 12:40	4 KB
file1.txt	-rw-rw----	2024-11-06 12:40	5 B