**Palindrome\PalindromeCheck.txt**

```Assignment 9 : Palindrome Number Check```

1. activity_palindrome_check.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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".palindrome_check">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="16dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="66dp"
            android:orientation="vertical">

            <TextView
                android:id="@+id/textView3"
                android:layout_width="match_parent"
                android:layout_height="64dp"
                android:background="@drawable/shape_btn"
                android:fontFamily="sans-serif-smallcaps"
                android:gravity="center"
                android:text="Palindrome Check"
                android:textSize="40dp"
                android:textStyle="bold"
                android:textColor="@color/black"/>

        </LinearLayout>

        <Space
            android:layout_width="match_parent"
            android:layout_height="103dp" />

        <EditText
        android:id="@+id/inputNumber1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter first number"
        android:inputType="number"
        android:layout_marginBottom="8dp" />

    <EditText
        android:id="@+id/inputNumber2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter second number"
        android:inputType="number"
        android:layout_marginBottom="16dp" />
```

```
    <Button
        android:id="@+id/checkButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Check Palindromes"
        android:layout_marginBottom="16dp" />

    <TextView
        android:id="@+id/resultText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="8dp"
        android:text="Results will be displayed here."
        android:textSize="16sp"
        android:textColor="@android:color/white" />

    </LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

2. palindrome_check.java :

```
package com.example.assignmenthub;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class palindrome_check extends AppCompatActivity {

    // Declare views
    private EditText inputNumber1, inputNumber2;
    private Button checkButton;
    private TextView resultText;

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

        // Initialize views
        inputNumber1 = findViewById(R.id.inputNumber1);
        inputNumber2 = findViewById(R.id.inputNumber2);
        checkButton = findViewById(R.id.checkButton);
        resultText = findViewById(R.id.resultText);

        // Set button click listener
        checkButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                palindromeCheck();
            }
        });
    }
```

```java
    // Method to handle palindrome check
    private void palindromeCheck() {
        // Get input numbers
        String num1String = inputNumber1.getText().toString();
        String num2String = inputNumber2.getText().toString();

        // Validate inputs
        if (num1String.isEmpty() || num2String.isEmpty()) {
            Toast.makeText(this, "Please enter both numbers", Toast.LENGTH_SHORT).show();
            return;
        }

        try {
            int num1 = Integer.parseInt(num1String);
            int num2 = Integer.parseInt(num2String);

            // Check for palindromes
            String result = palindromeCheckFun(num1, num2);
            resultText.setText(result);
        } catch (NumberFormatException e) {
            Toast.makeText(this, "Please enter valid numbers", Toast.LENGTH_SHORT).show();
        }
    }

    // Method to check if two numbers are palindromes
    private String palindromeCheckFun(int num1, int num2) {
        boolean isNum1Palindrome = isPalindrome(num1);
        boolean isNum2Palindrome = isPalindrome(num2);

        // Construct the result string
        StringBuilder result = new StringBuilder();
        result.append(num1).append(isNum1Palindrome ? " is a palindrome.\n" : " is not a
palindrome.\n");
        result.append(num2).append(isNum2Palindrome ? " is a palindrome." : " is not a
palindrome.");

        return result.toString();
    }

    // Function to check if a number is a palindrome
    private boolean isPalindrome(int number) {
        if (number < 0) return false; // Negative numbers are not palindromes
        int original = number;
        int reversed = 0;

        while (number > 0) {
            int digit = number % 10;
            reversed = reversed * 10 + digit;
            number /= 10;
        }

        return original == reversed;
    }
}
```