

# Báo cáo bài tập về nhà ngày 4/11/2025

Họ và tên sinh viên: Phùng Duy Nghĩa

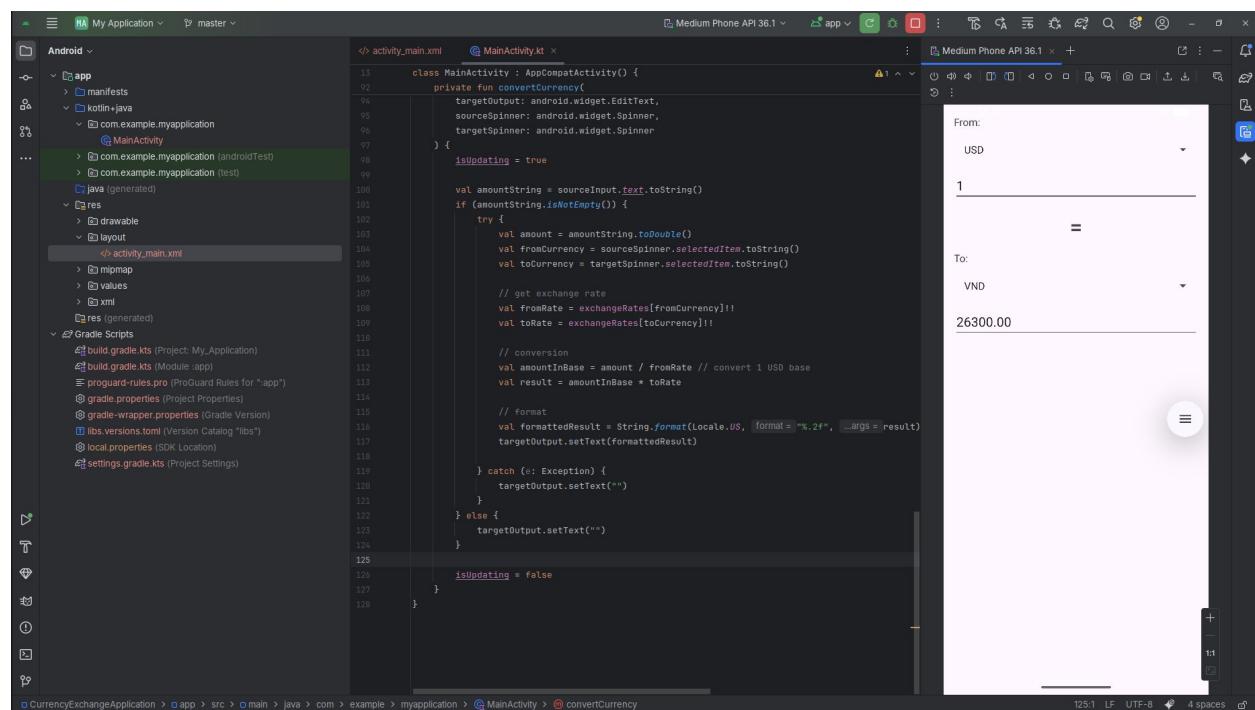
MSSV: 20225896

Link github các bài tập: <https://github.com/nghiaphunng18/mobile-programming>

BT1: Lập trình ứng dụng đổi tiền tệ

Link github: <https://github.com/nghiaphunng18/mobile-programming/tree/master/CurrencyExchangeApplication>

1. Demo USD -> VND



```
13     class MainActivity : AppCompatActivity() {
14         private fun convertCurrency(
15             targetOutput: android.widget.EditText,
16             sourceSpinner: android.widget.Spinner,
17             targetSpinner: android.widget.Spinner
18         ) {
19             isUpdating = true
20
21             val amountString = sourceInput.text.toString()
22             if (amountString.isNotEmpty()) {
23                 try {
24                     val amount = amountString.toDouble()
25                     val fromCurrency = sourceSpinner.selectedItem.toString()
26                     val toCurrency = targetSpinner.selectedItem.toString()
27
28                     // get exchange rate
29                     val fromRate = exchangeRates[fromCurrency]!!
30                     val toRate = exchangeRates[toCurrency]!!
31
32                     // conversion
33                     val amountInBase = amount / fromRate // convert 1 USD base
34                     val result = amountInBase * toRate
35
36                     // format
37                     val formattedResult = String.format(Locale.US, "%,.2f", result)
38                     targetOutput.setText(formattedResult)
39
40                 } catch (e: Exception) {
41                     targetOutput.setText("")
42                 }
43             } else {
44                 targetOutput.setText("")
45             }
46             isUpdating = false
47         }
48     }
```

2. Demo JPY -> VND

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "My Application".
- MainActivity.kt:** The main code editor displays the Java code for `MainActivity`. The code handles currency conversion between JPY and VND.
- UI Preview:** On the right, there is a preview window titled "Medium Phone API 36.1" showing a currency conversion form. It has fields for "From" (JPY) and "To" (VND), and a result field showing "171.45".
- Status Bar:** At the bottom, it shows "125:1 LF UFT-8 4 spaces".

BT2: Lập trình ứng dụng hiển thị danh sách số nguyên theo yêu cầu

Link github: <https://github.com/nghiaphunng18/mobile-programming/tree/master/IntegerListApplication>

### 1. Demo dãy số lẻ

The screenshot shows the Android Studio interface with the project structure on the left and two code editors on the right. The left editor shows `MainActivity.kt` and the right editor shows `NumberAdapter.kt`. A floating window is overlaid on the screen, displaying a sequence of numbers from 1 to 50. To the right of the sequence is a sidebar with several checkboxes:

- Số lẻ (Odd number) - checked
- Số nguyên tố (Prime number)
- Số hoàn hảo (Perfect number)
- Số chẵn (Even number)
- Số chính phương (Perfect square)
- Số Fibonacci

## 2. Demo dãy số nguyên tố

The screenshot shows the Android Studio interface with the project structure on the left and two code editors on the right. The left editor shows `MainActivity.kt` and the right editor shows `NumberAdapter.kt`. A floating window is overlaid on the screen, displaying a sequence of numbers from 1 to 50. To the right of the sequence is a sidebar with several checkboxes:

- Số lẻ (Odd number)
- Số nguyên tố (Prime number) - checked
- Số hoàn hảo (Perfect number)
- Số chẵn (Even number)
- Số chính phương (Perfect square)
- Số Fibonacci

## 3. Demo dãy số hoàn hảo

The screenshot shows the Android Studio interface with the project structure on the left and the code editor on the right. The code editor displays the `MainActivity.kt` file, which contains logic for generating numbers based on user input. A floating search bar is visible on the right side of the screen, containing the number 50. Below the search bar, there are several radio buttons for filtering results: Số lẻ, Số nguyên tố, Số hoàn hảo, Số chẵn, Số chính phương, and Số Fibonacci.

```

13     class MainActivity : AppCompatActivity() {
14         private fun updateList() {
15             R.id.rbPerfect -> isPerfectNumber(n = 1)
16             R.id.rbFibonacci -> isFibonacci(n = 1)
17             else -> false
18         }
19         if (shouldAdd) {
20             generatedNumbers.add(1)
21         }
22
23         if (generatedNumbers.isEmpty()) {
24             binding.recyclerView.visibility = View.GONE
25             binding.tvEmpty.visibility = View.VISIBLE
26         } else {
27             binding.recyclerView.visibility = View.VISIBLE
28             binding.tvEmpty.visibility = View.GONE
29             numberAdapter.submitList(newNumbers = generatedNumbers)
30         }
31     }
32
33     1Usage
34     private fun isOdd(n: Int): Boolean = n % 2 != 0
35     1Usage
36     private fun isEven(n: Int): Boolean = n % 2 == 0
37     1Usage
38     private fun isPrime(n: Int): Boolean { ... }
39     3 Usages
40     private fun isPerfectSquare(n: Long): Boolean { ... }
41     1Usage
42     private fun isPerfectNumber(n: Int): Boolean { ... }
43     1Usage
44     private fun isFibonacci(n: Int): Boolean {
45         val nLong = n.toLong()
46         return isPerfectSquare(n = 5L * nLong * nLong + 4) || isPerfectSquare(n = 4L * nLong * nLong + 1)
47     }
48
49 }

```

#### 4. Demo dãy số chẵn

This screenshot is similar to the previous one, but the floating search bar has the number 50 selected. The radio button for "Số chẵn" is checked, indicating that the search results will be filtered to show even numbers. The rest of the UI and code are identical to the first screenshot.

#### 5. Demo dãy số chính phương

The screenshot shows the Android Studio interface with the project structure on the left and two code editors on the right. The top editor contains the `MainActivity.kt` file, which includes logic for generating numbers based on user input and displaying them in a RecyclerView. The bottom editor contains the `NumberAdapter.kt` file, which handles the adapter for the RecyclerView. A floating window on the right displays a sequence of numbers from 1 to 50, with each number having a circular checkbox next to it. The checkboxes are labeled: Số lẻ (Odd), Số nguyên tố (Prime), Số hoàn hảo (Perfect), Số chẵn (Even), Số chính phương (Square), and Sô Fibonacci (Fibonacci). The checkboxes for odd numbers, prime numbers, and perfect numbers are empty. The checkboxes for even numbers, square numbers, and Fibonacci numbers are filled.

```

class MainActivity : AppCompatActivity() {
    private fun updateList() {
        R.id.rbPerfect -> isPerfectNumber(n = 1)
        R.id.rbFibonacci -> isFibonacci(n = 1)
        else -> false
    }
    if (shouldAdd) {
        generatedNumbers.add(1)
    }

    if (generatedNumbers.isEmpty()) {
        binding.recyclerView.visibility = View.GONE
        binding.tvEmpty.visibility = View.VISIBLE
    } else {
        binding.recyclerView.visibility = View.VISIBLE
        binding.tvEmpty.visibility = View.GONE
        numberAdapter.submitList(newNumbers = generatedNumbers)
    }
}

private fun isOdd(n: Int): Boolean = n % 2 != 0
private fun isEven(n: Int): Boolean = n % 2 == 0
private fun isPrime(n: Int): Boolean = ...
private fun isPerfectSquare(n: Long): Boolean = ...
private fun isPerfectNumber(n: Int): Boolean = ...
private fun isFibonacci(n: Int): Boolean {
    val nLong = n.toLong()
    return isPerfectSquare(n = 5L * nLong * nLong + 4) || isPerfectSquare(n = 5L * nLong * nLong + 4)
}

```

## 6. Demo dãy số fibonacci

This screenshot is identical to the one above, showing the same code in `MainActivity.kt` and the same floating window displaying the first 50 numbers of the Fibonacci sequence. The checkboxes for odd numbers, prime numbers, and perfect numbers remain empty. The checkboxes for even numbers, square numbers, and Fibonacci numbers are filled.