Communciation System Lab’s Report  
Homework CH2



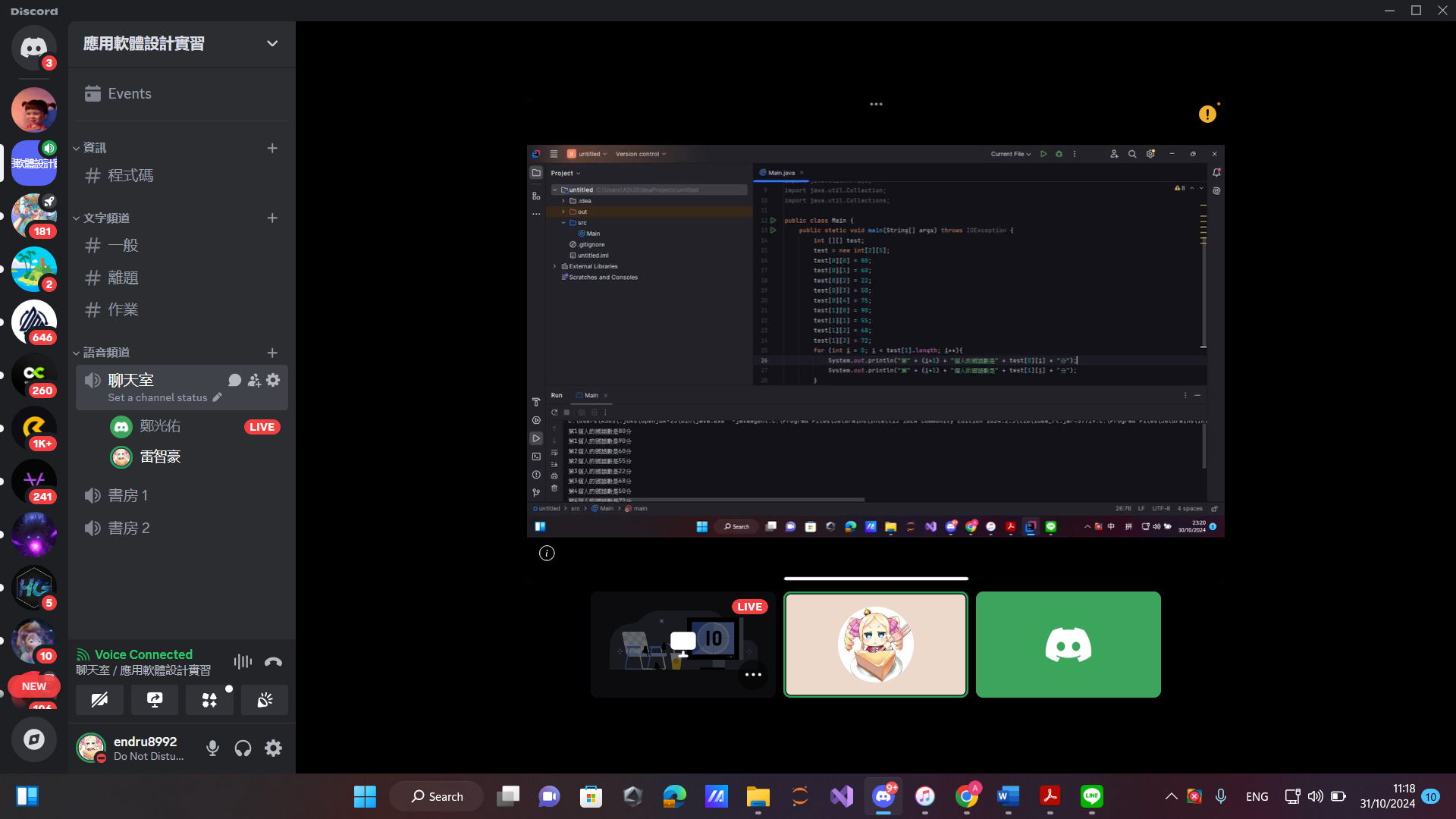
Name: 雷智豪

Class: 電子三甲

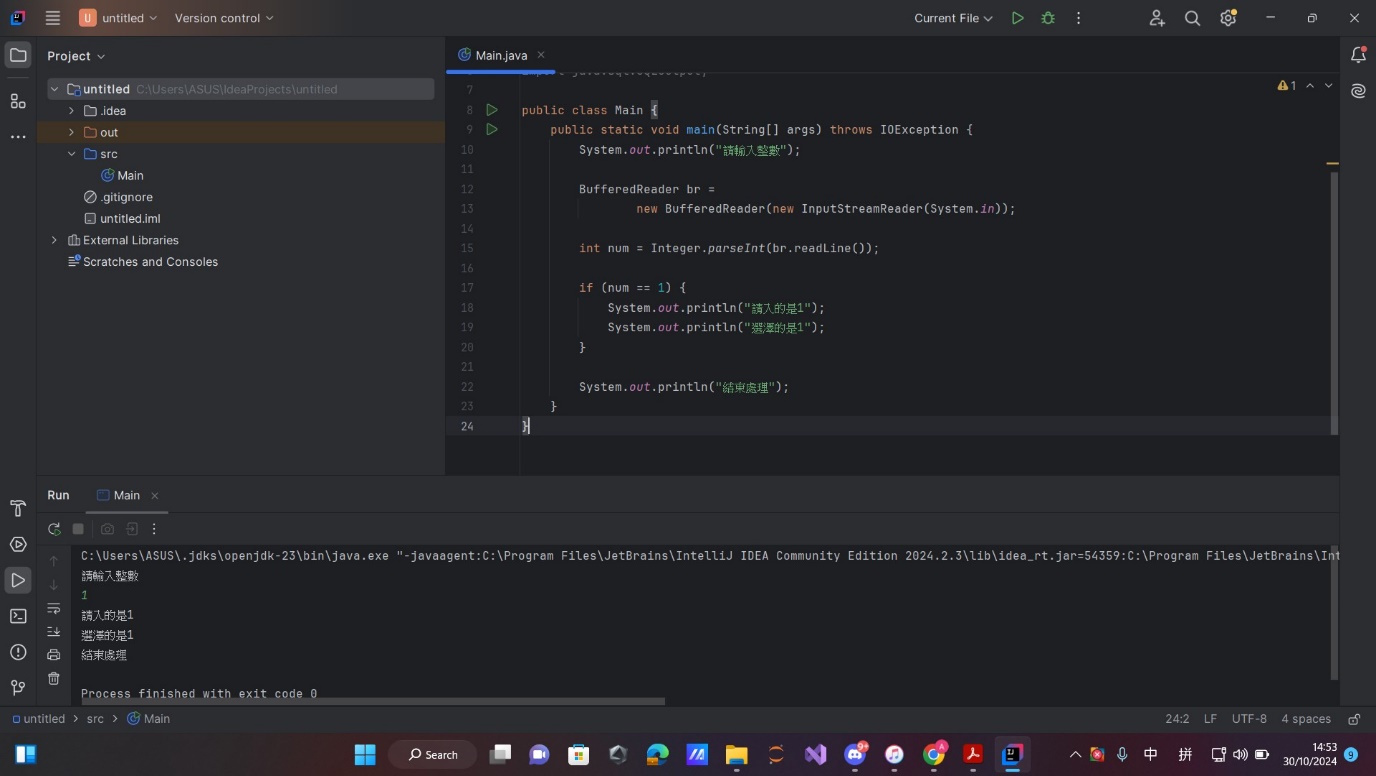
Student ID: 111360155

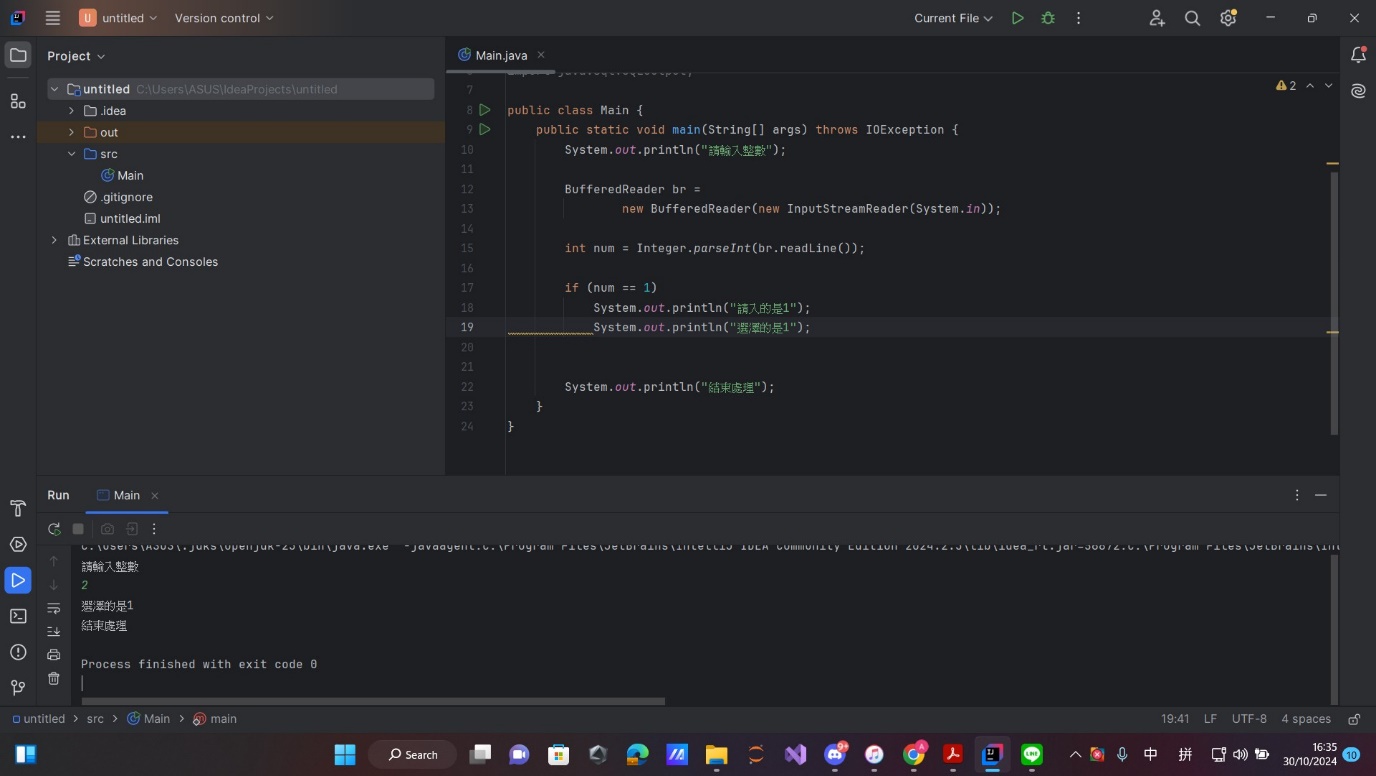
Group : 雷智豪 111360155

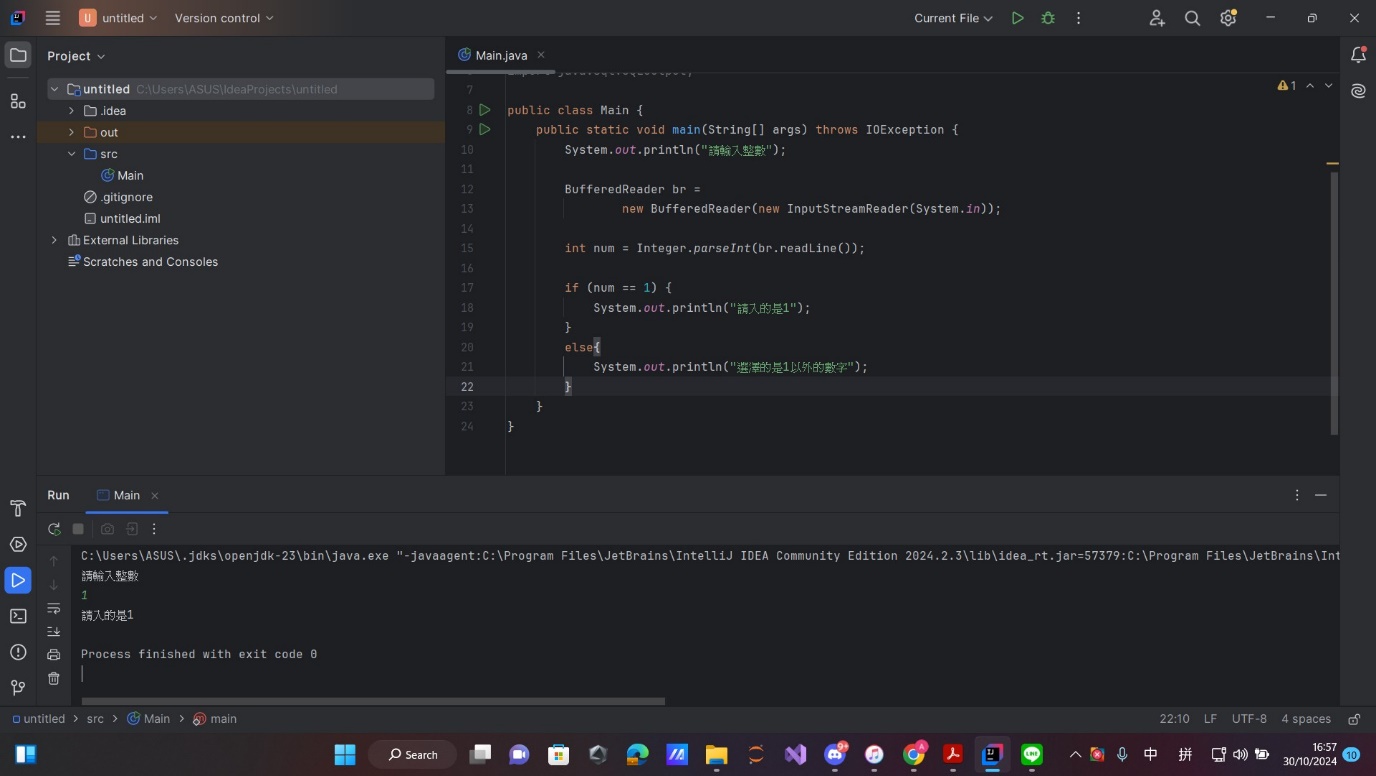
鄭光佑 109360105

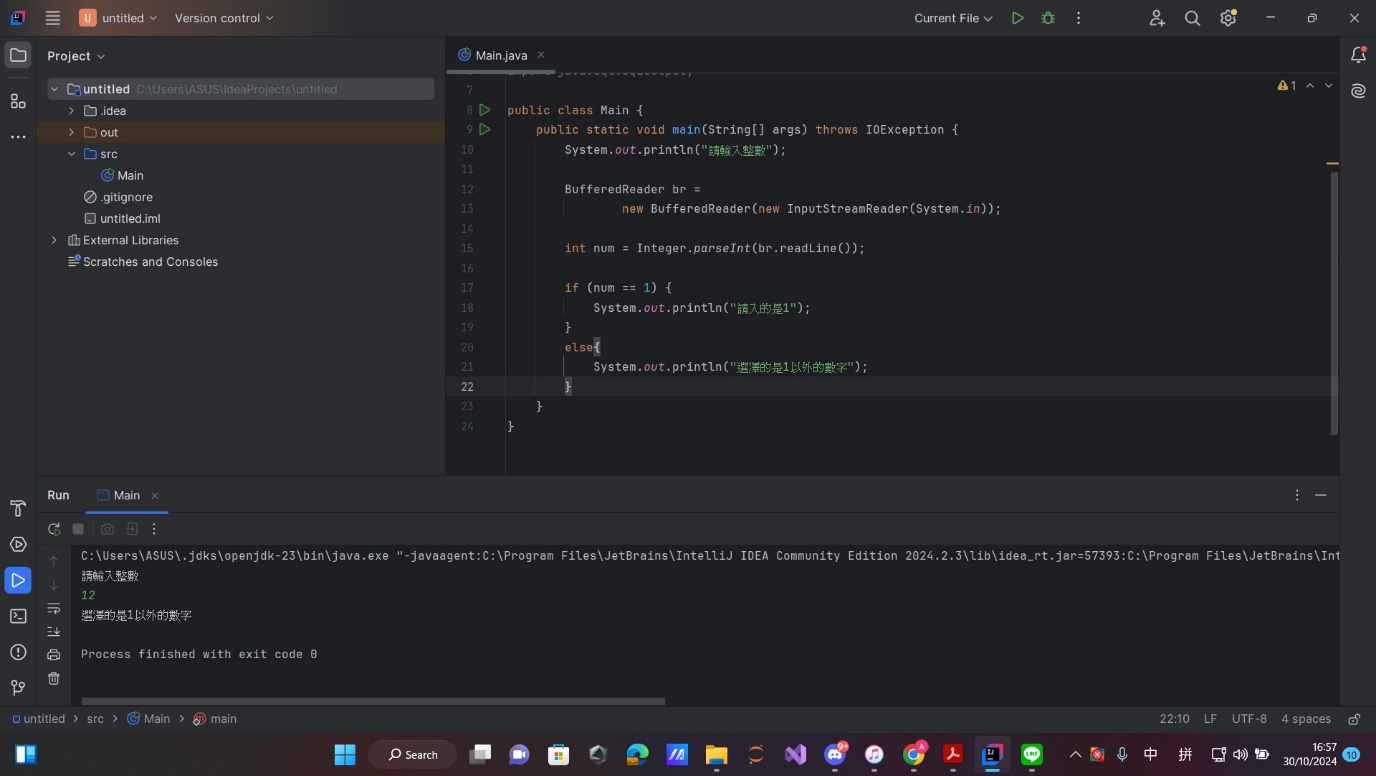


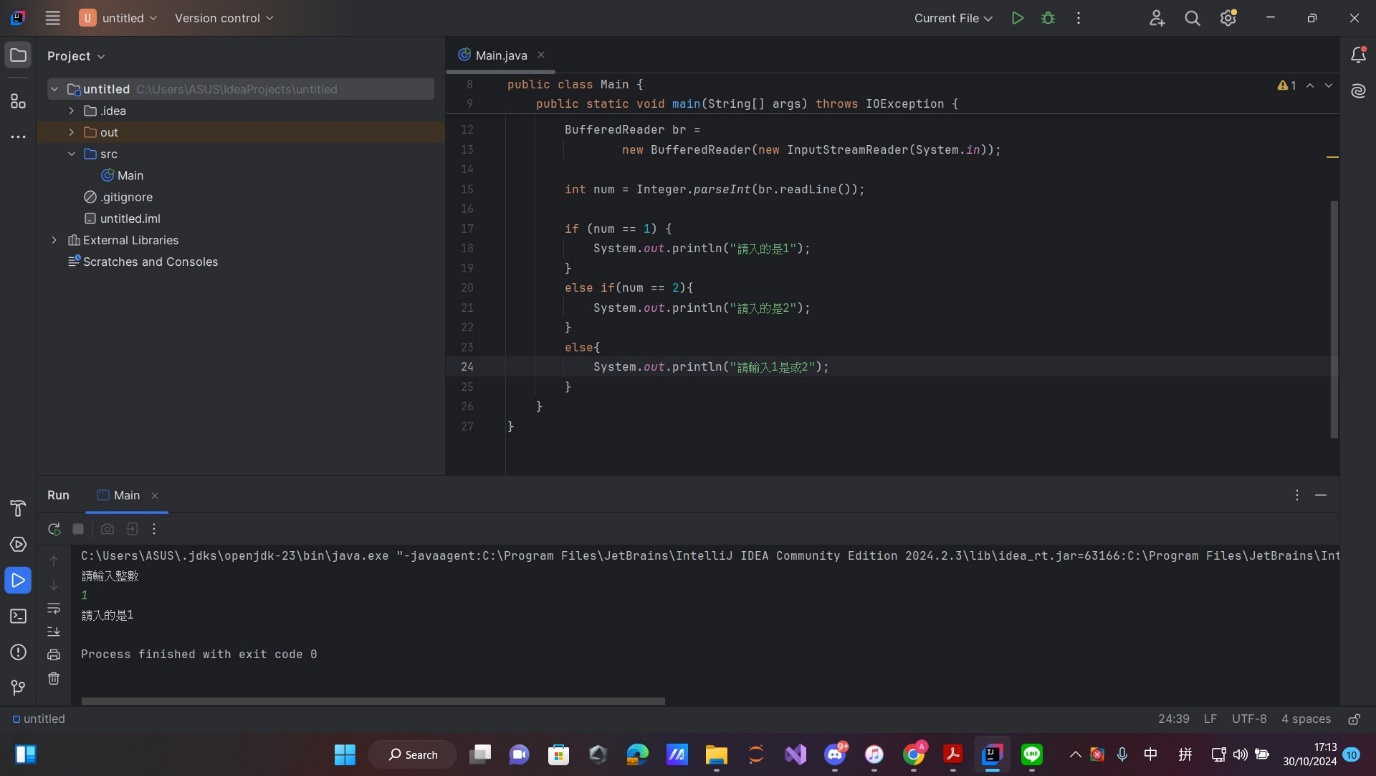
Java :

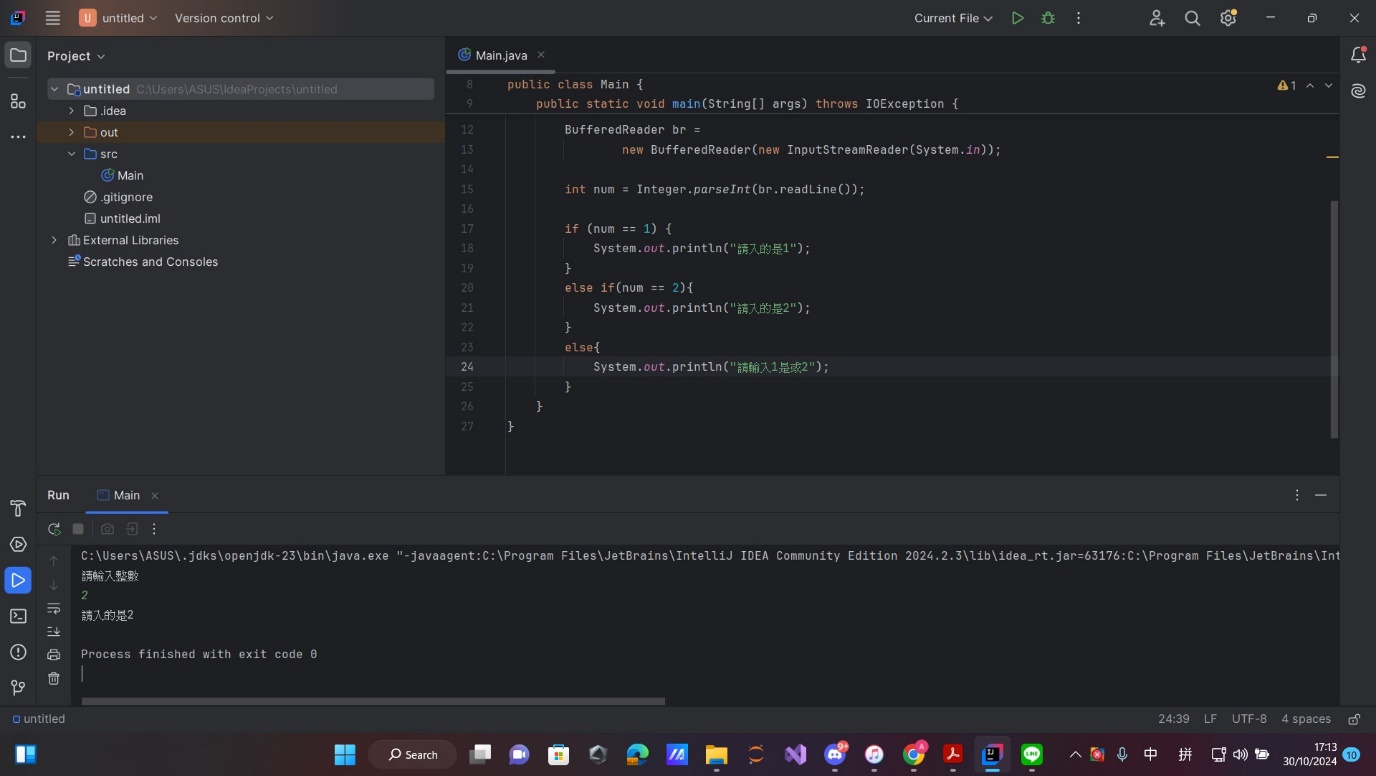


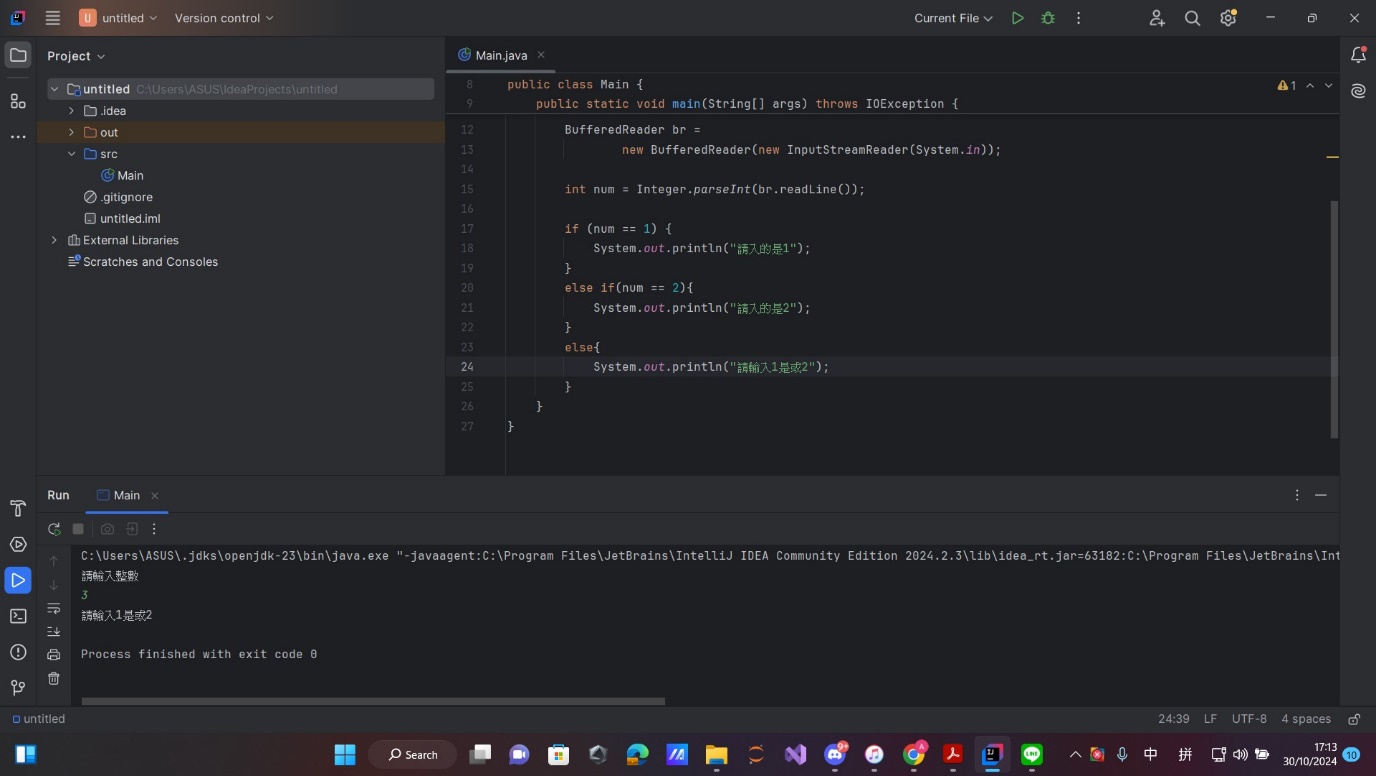


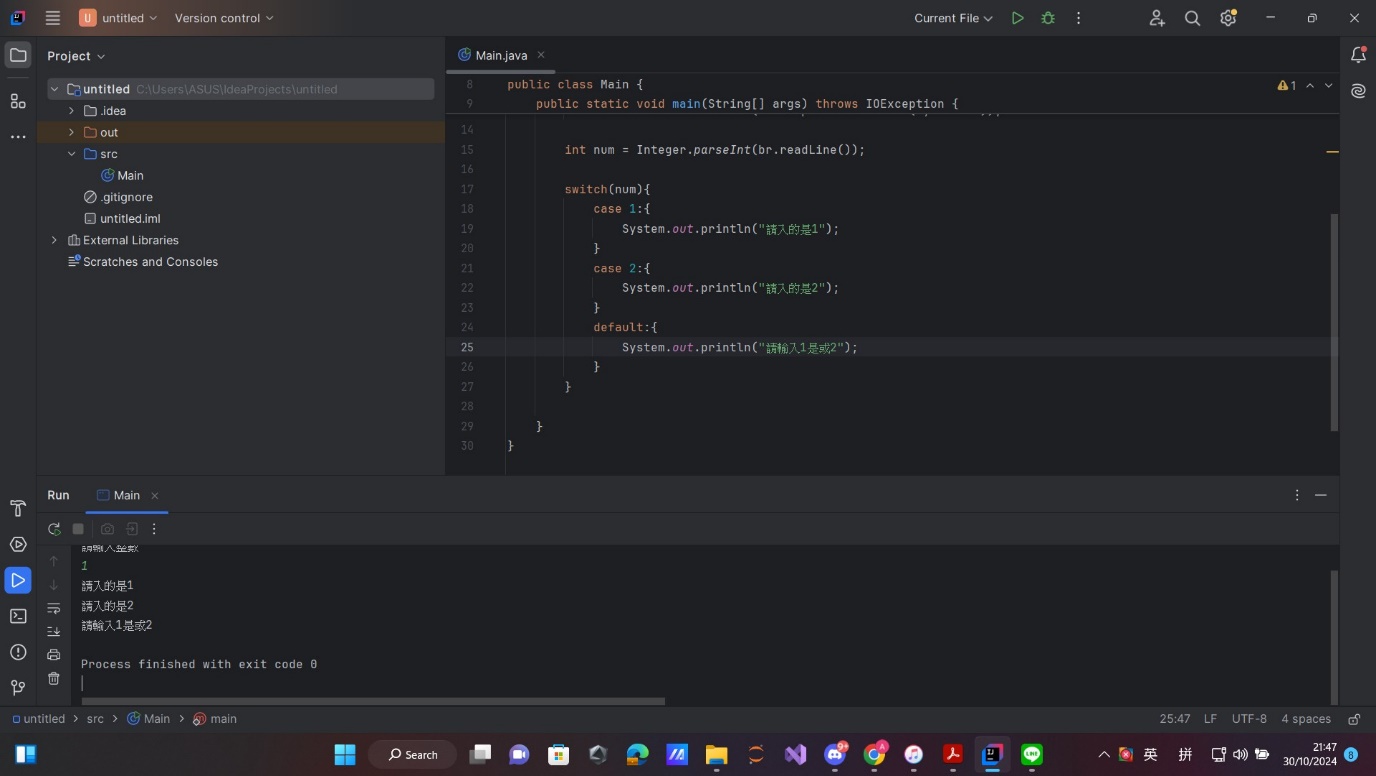


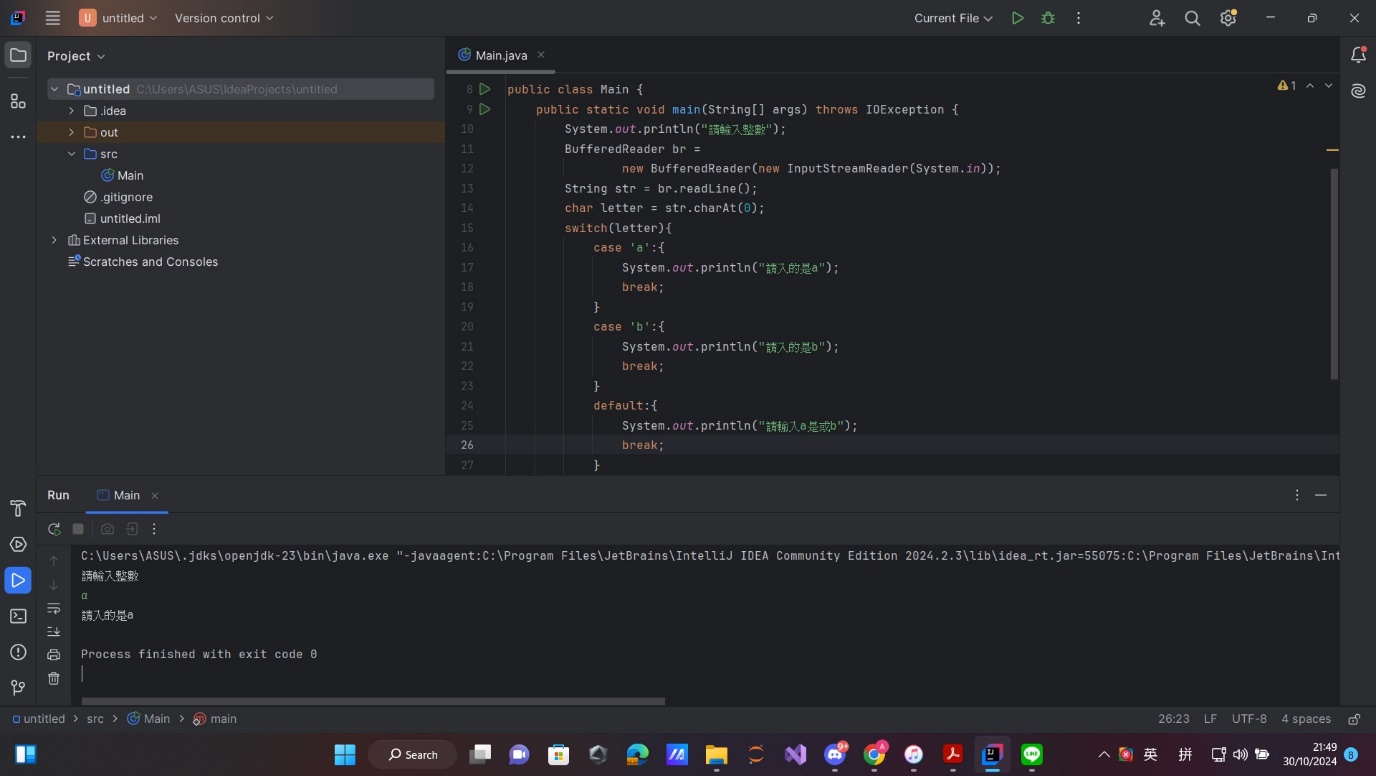


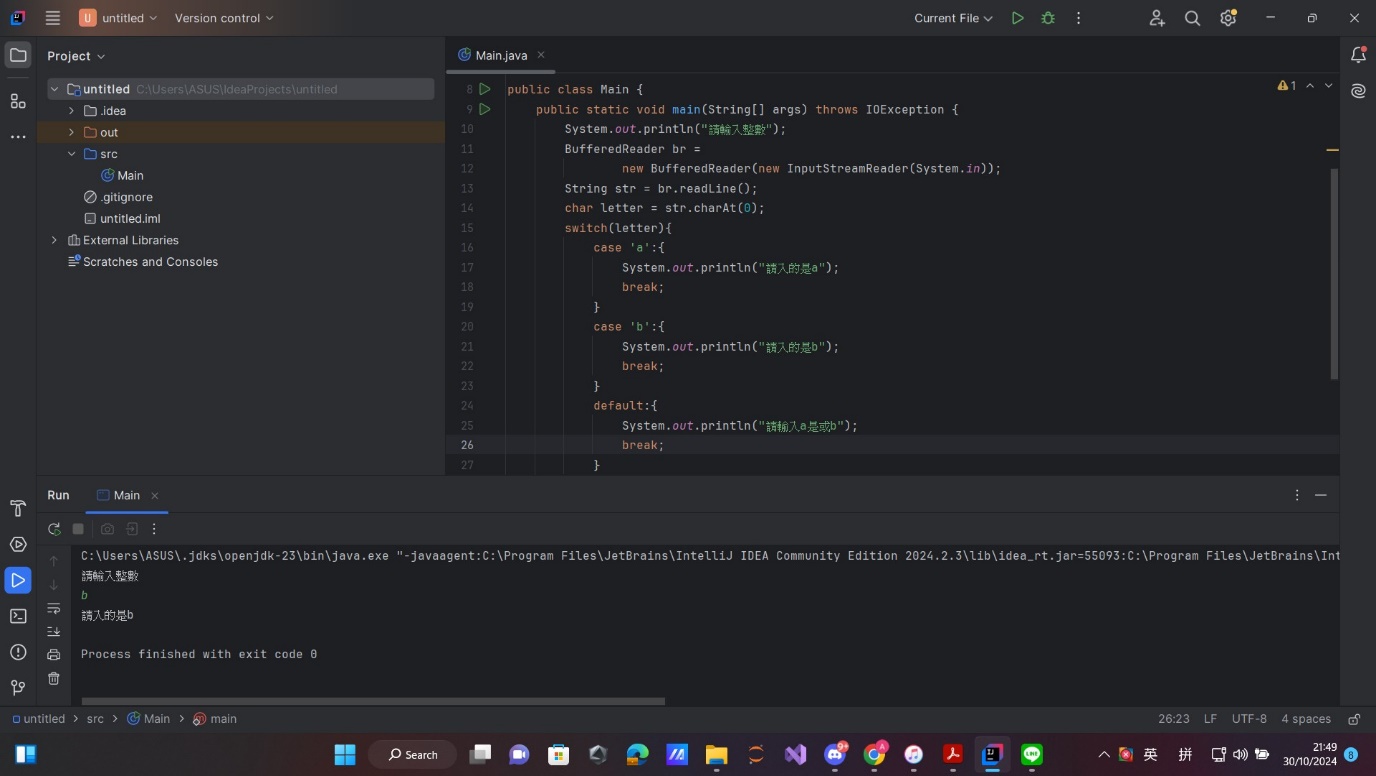


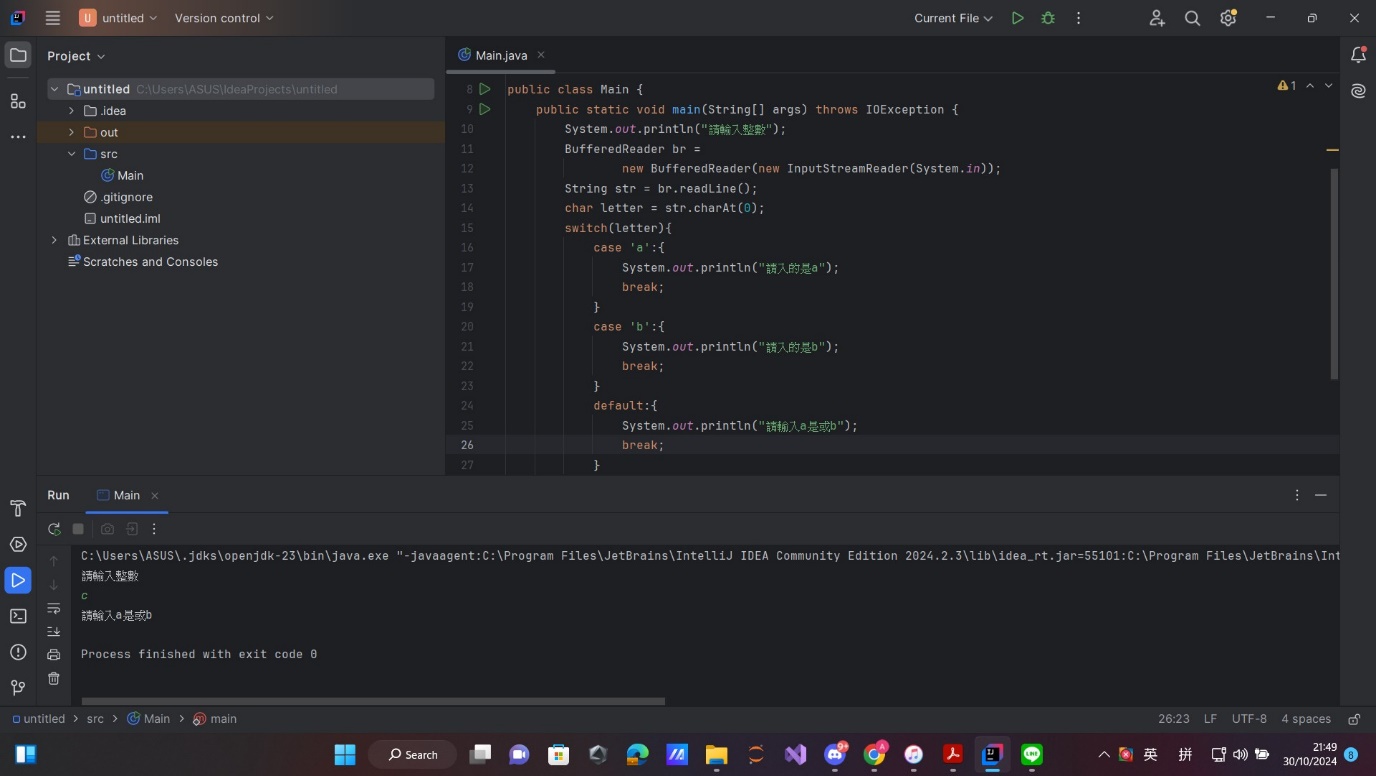


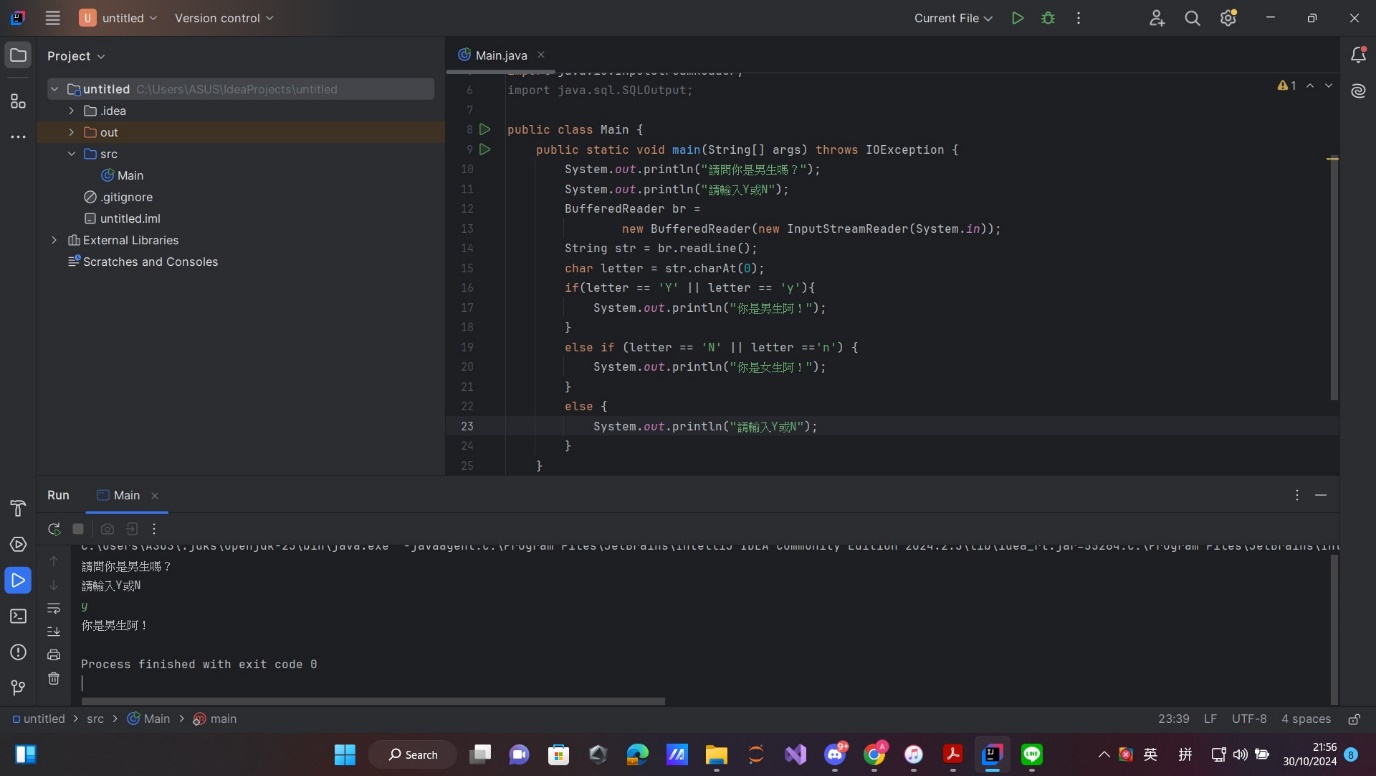


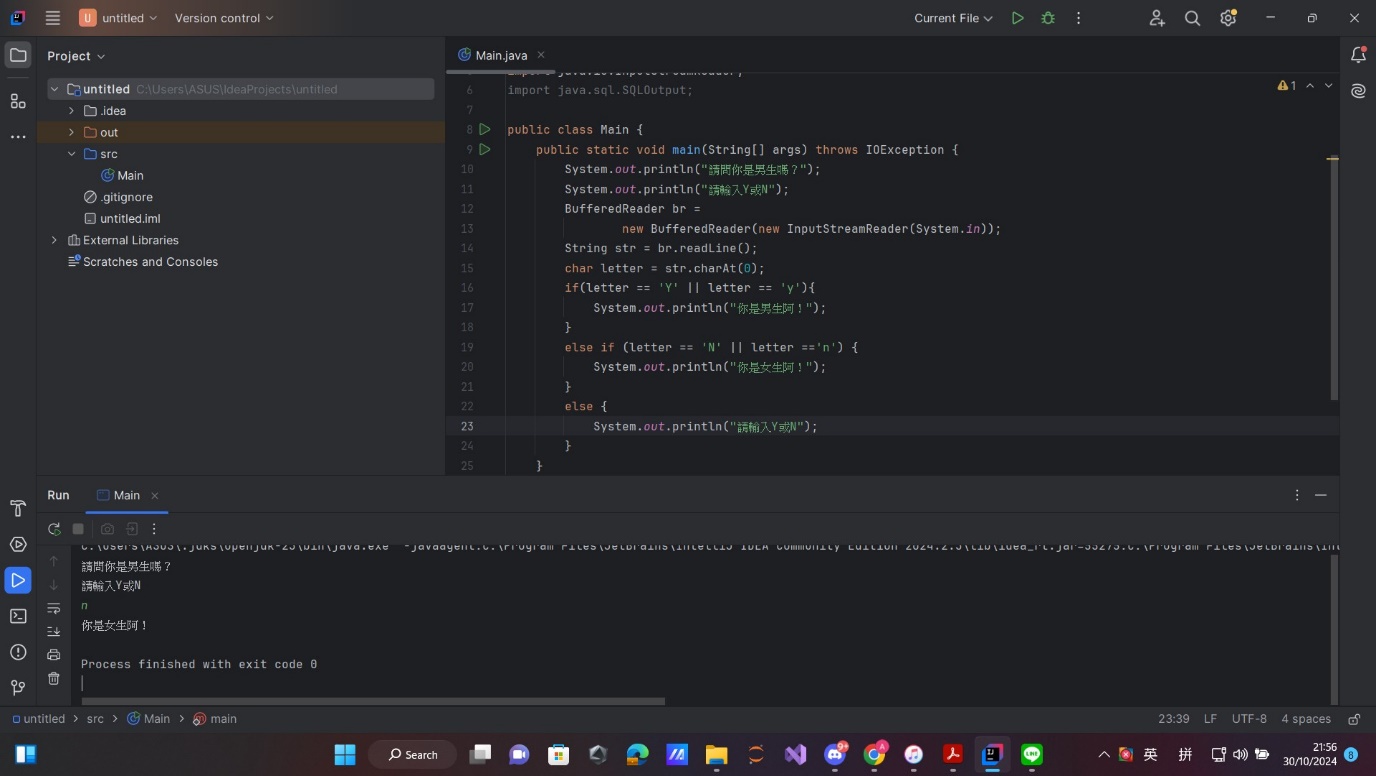


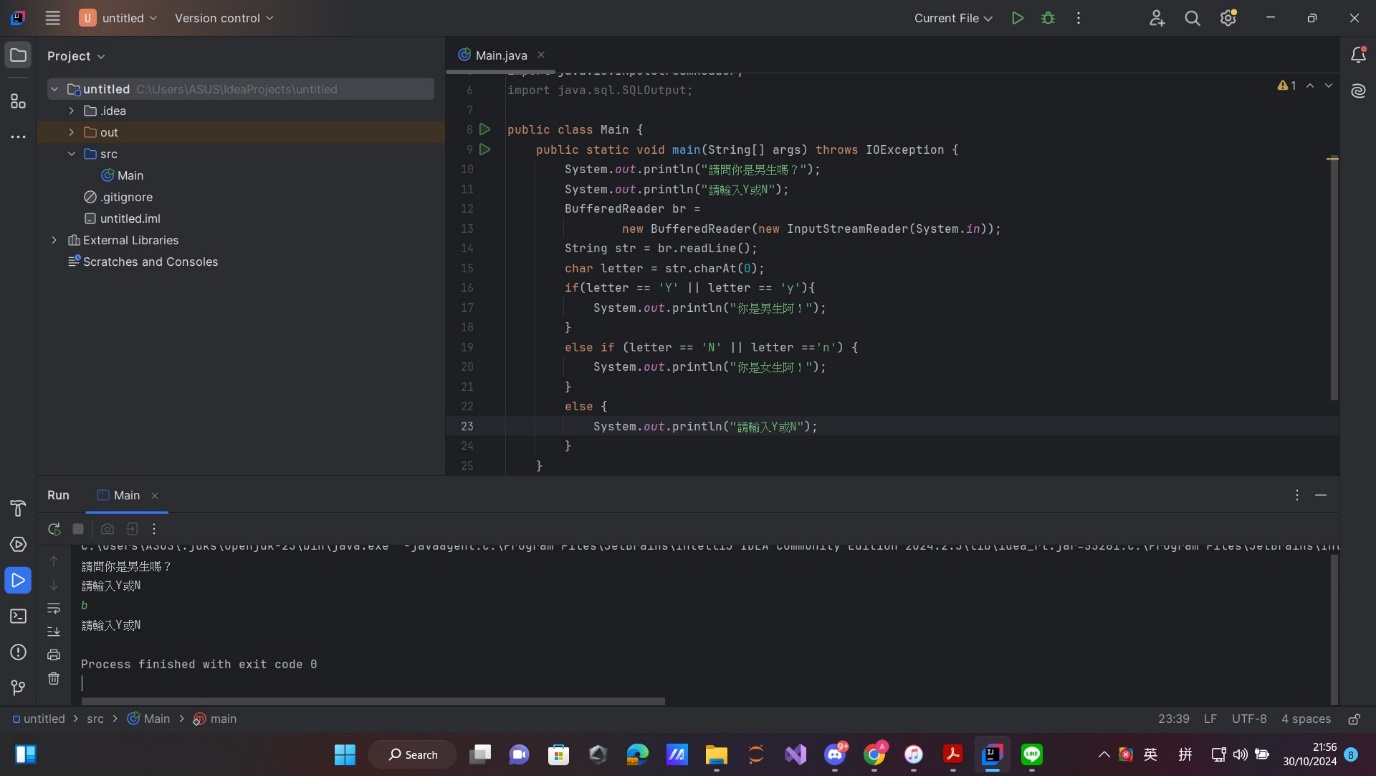


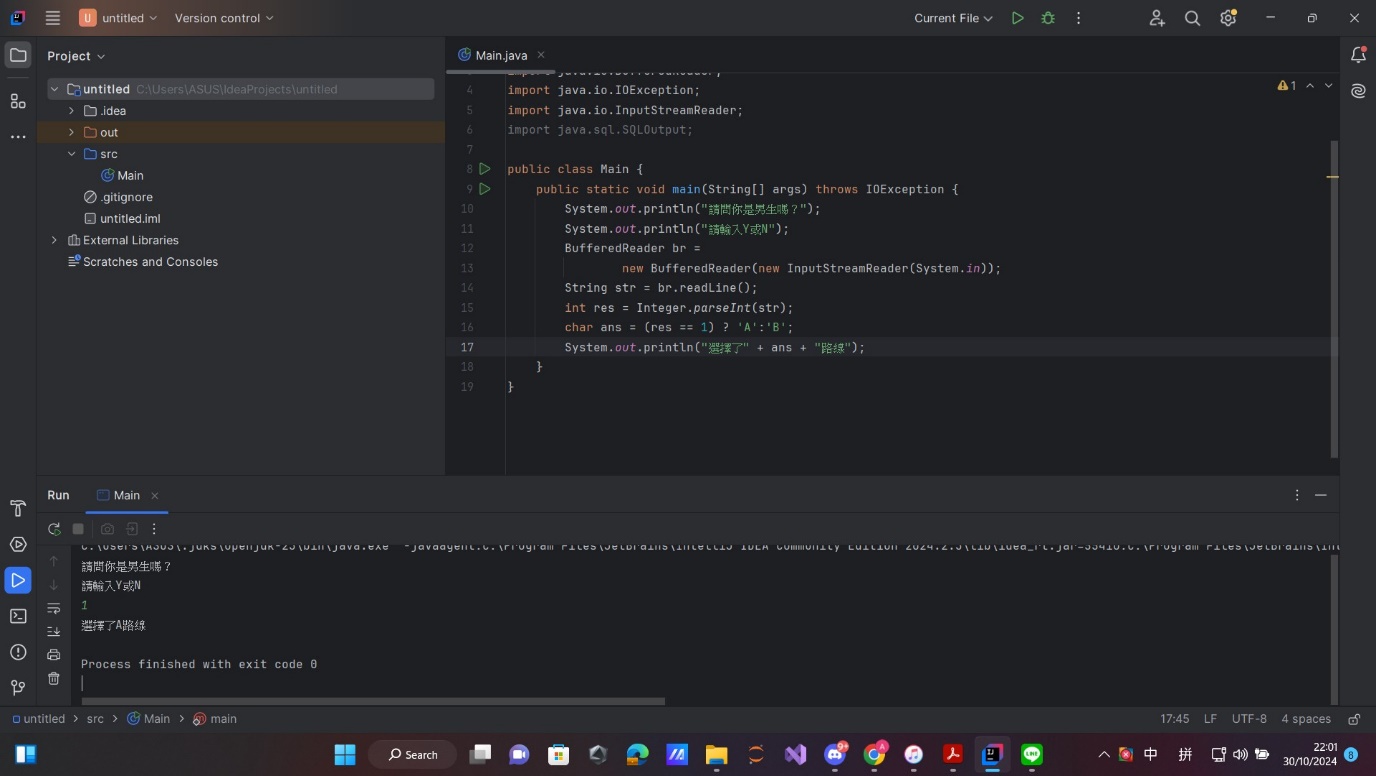


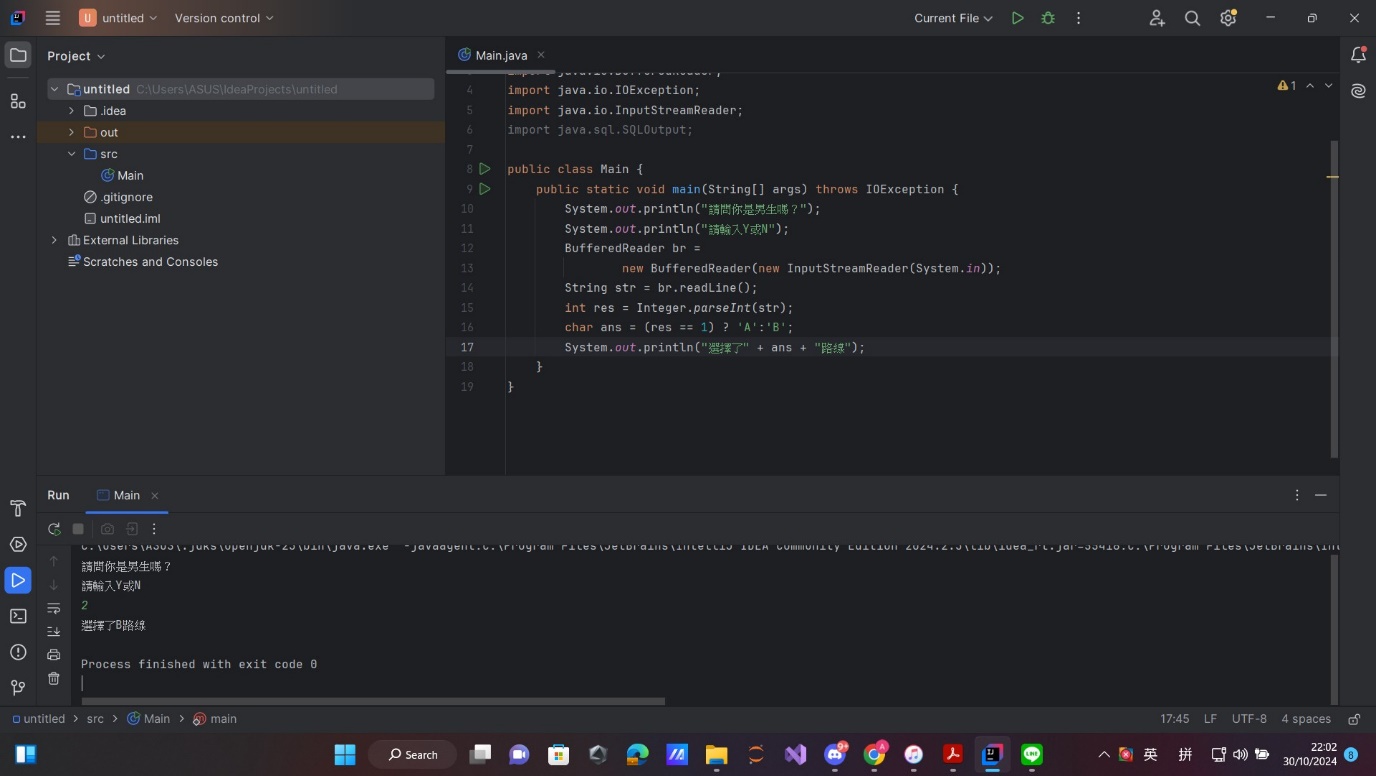


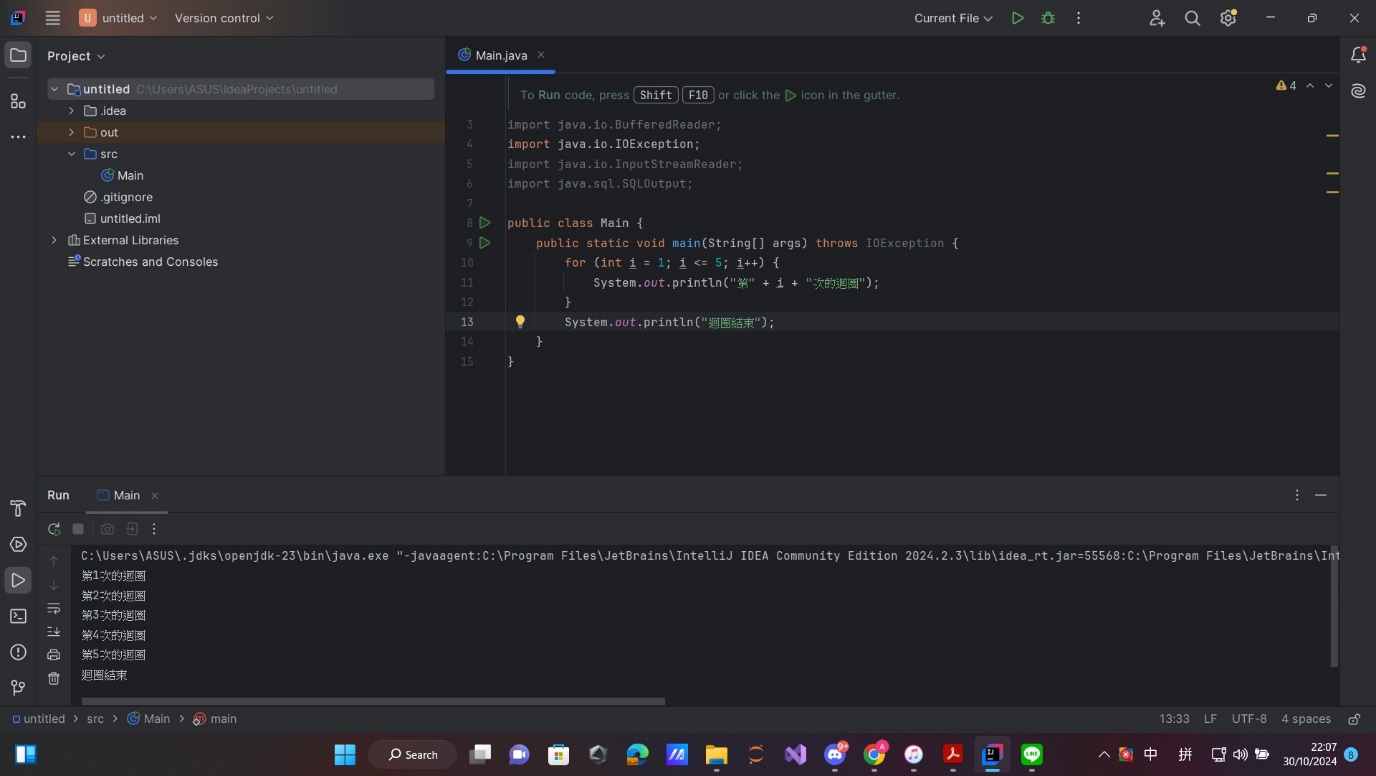


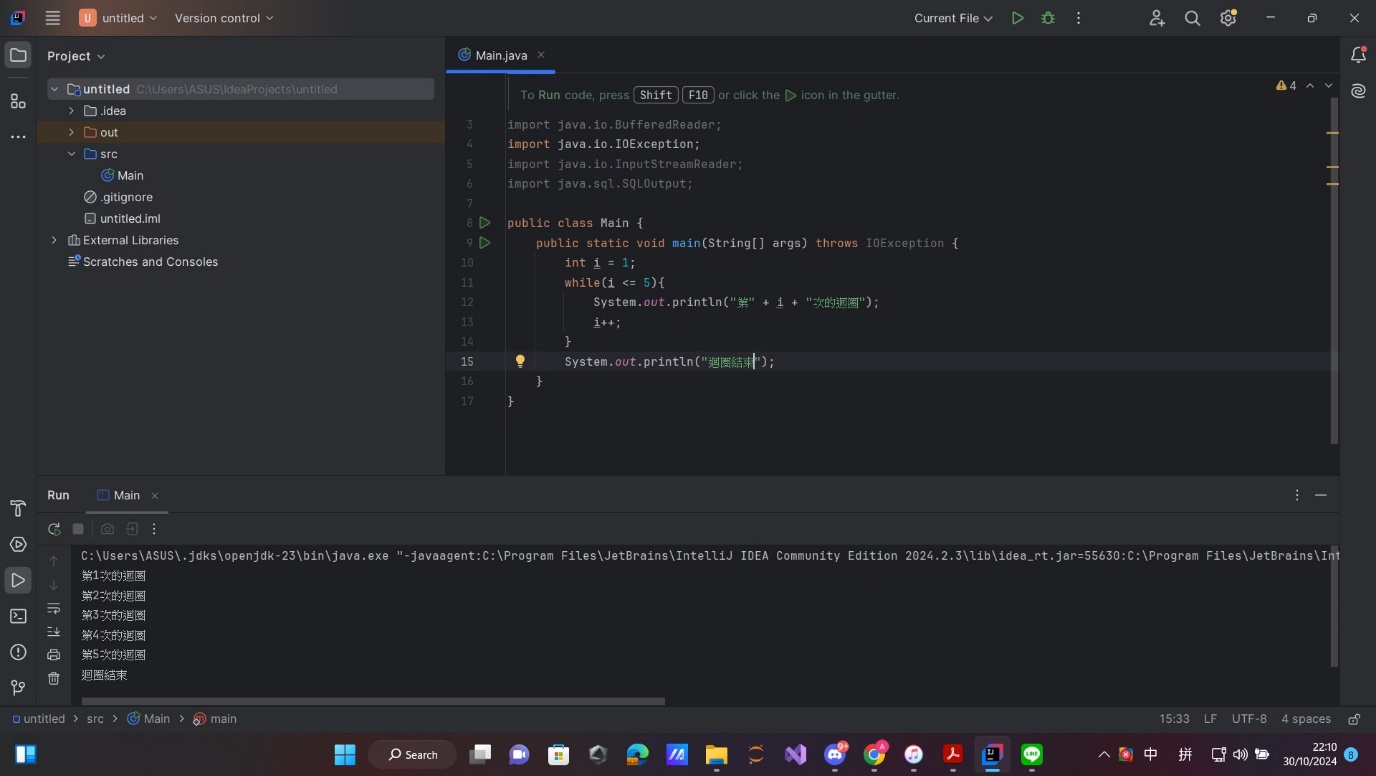


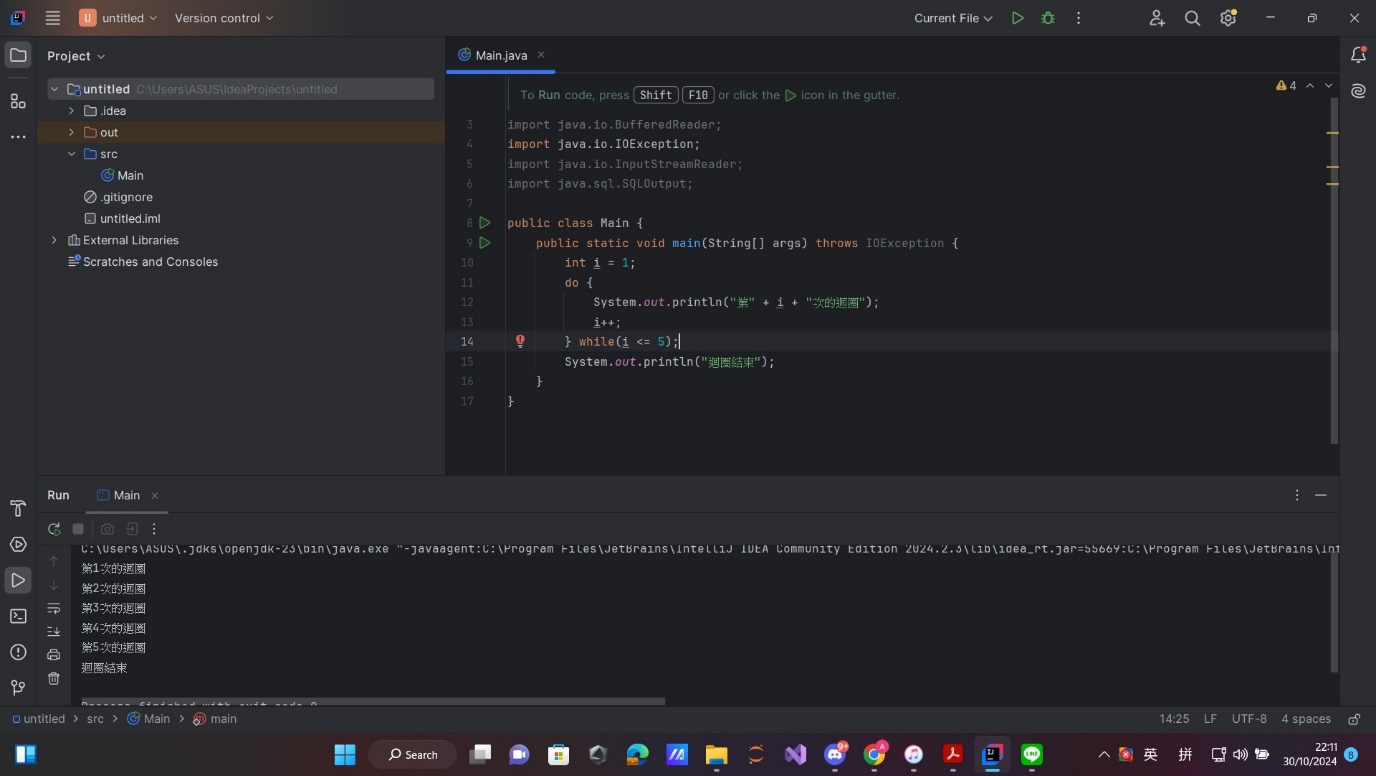


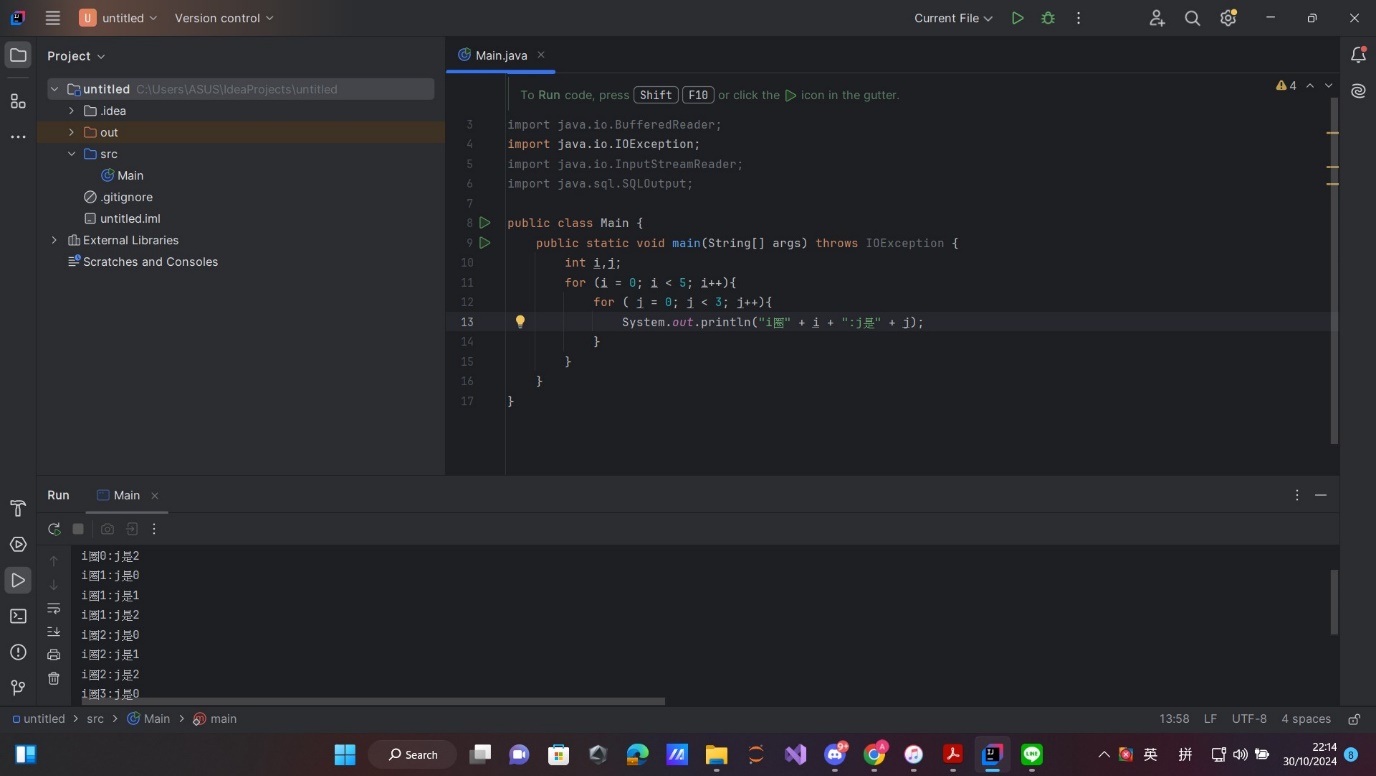


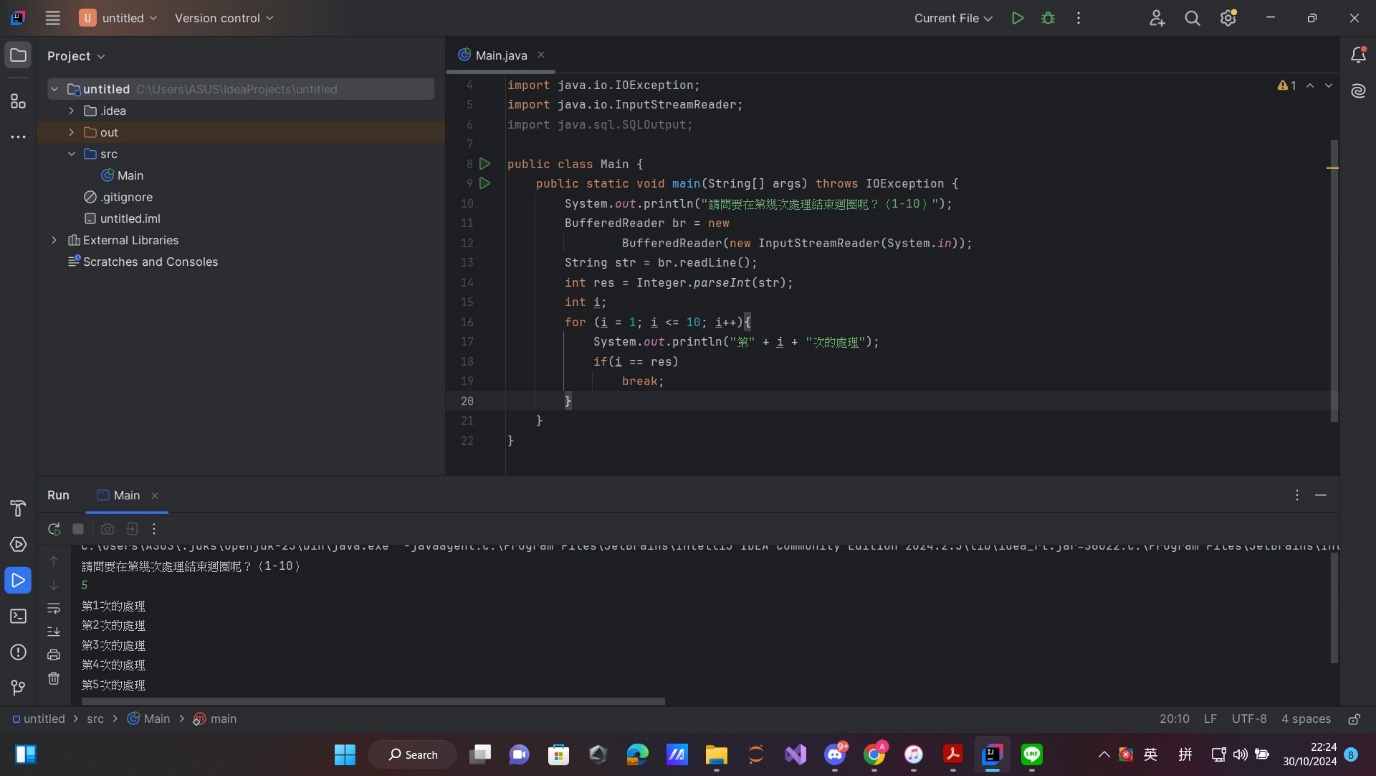


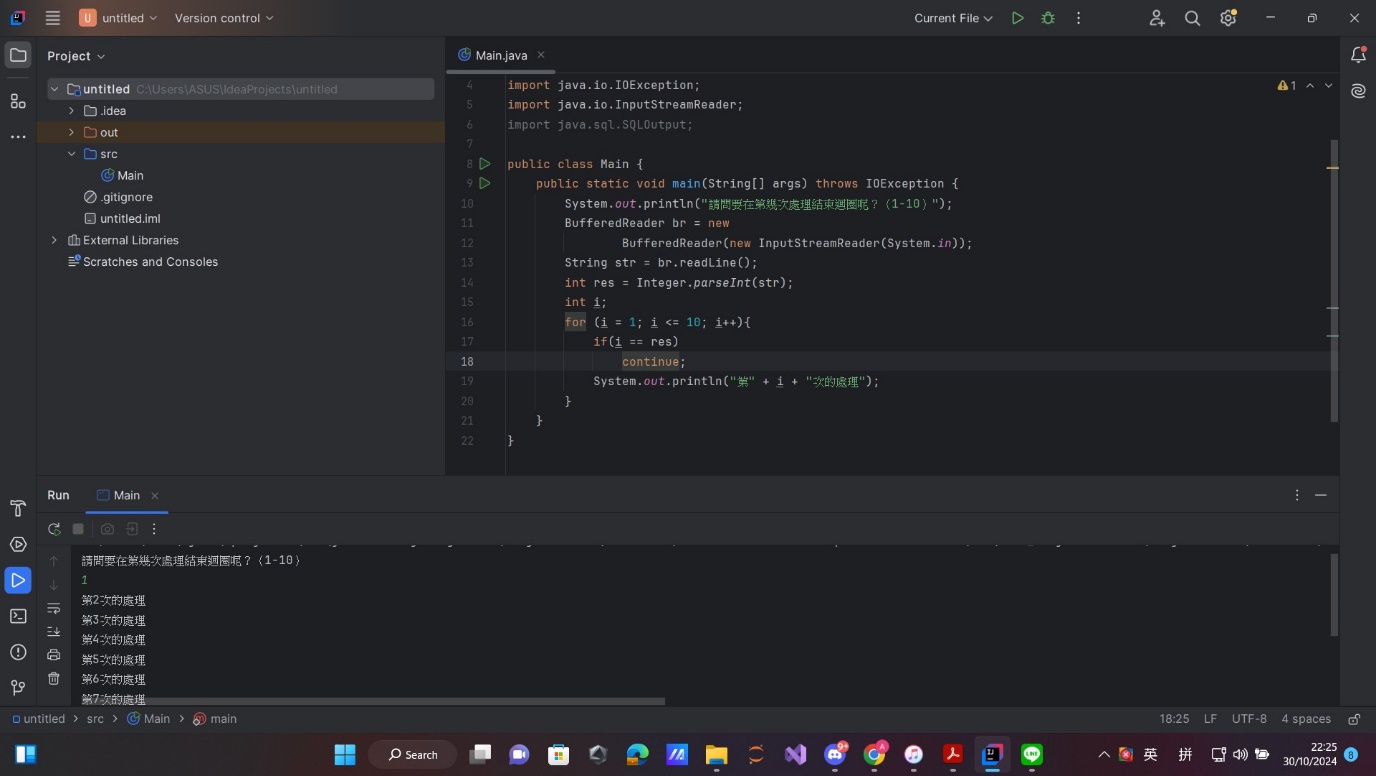


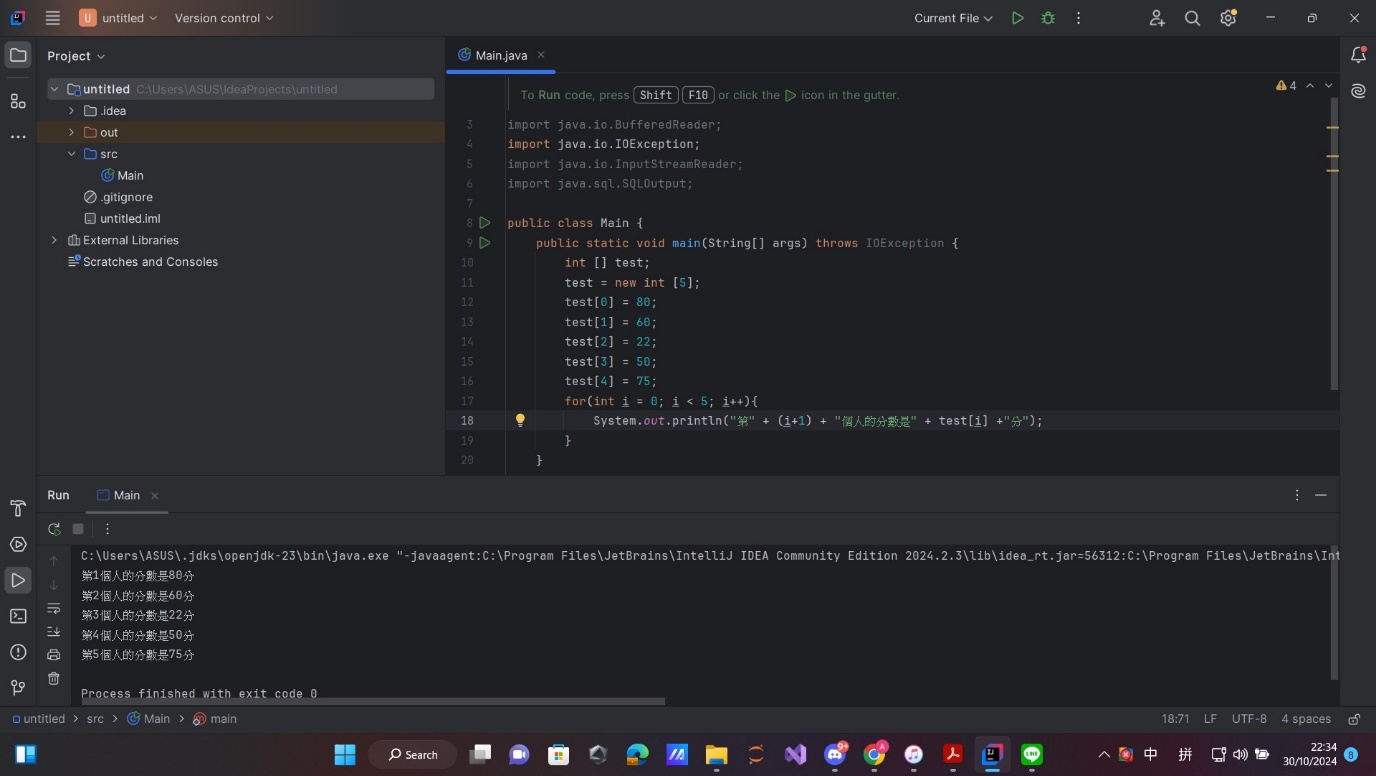


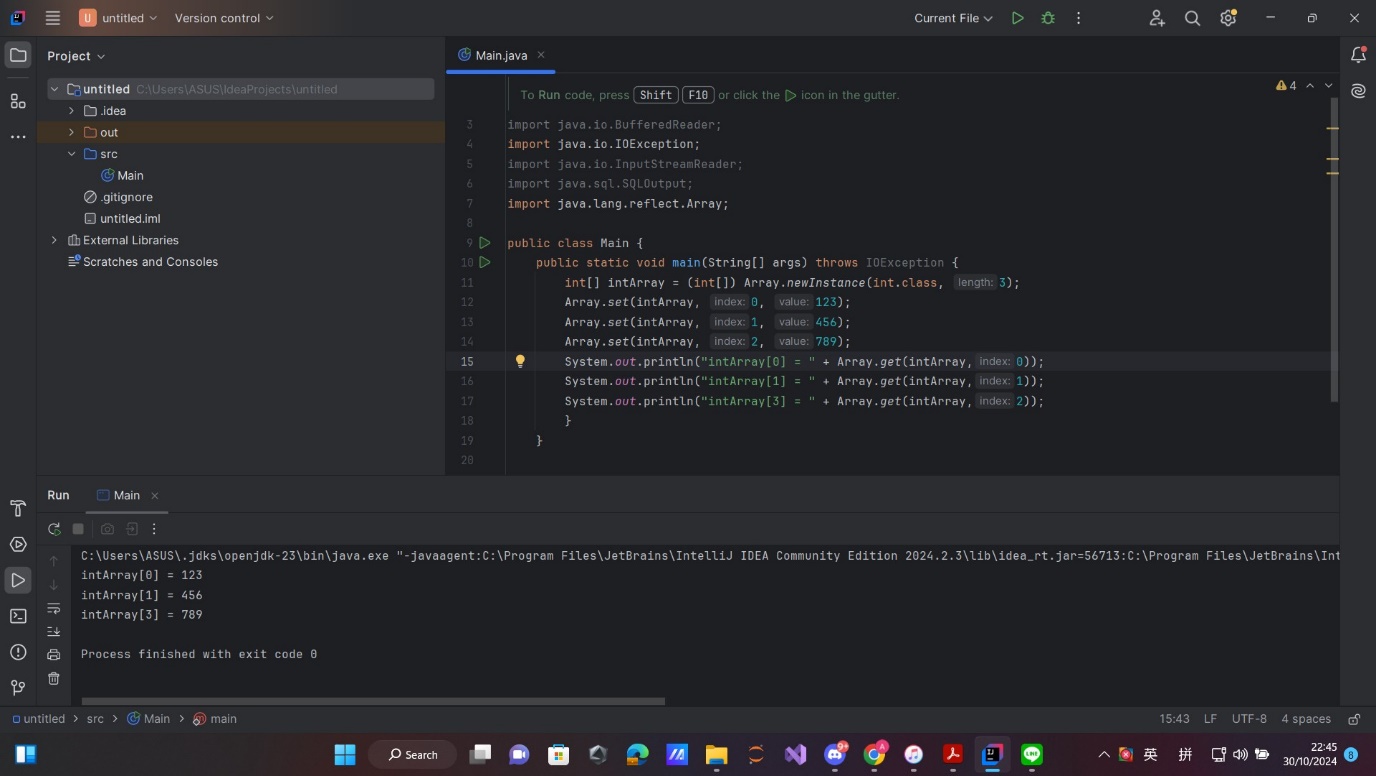


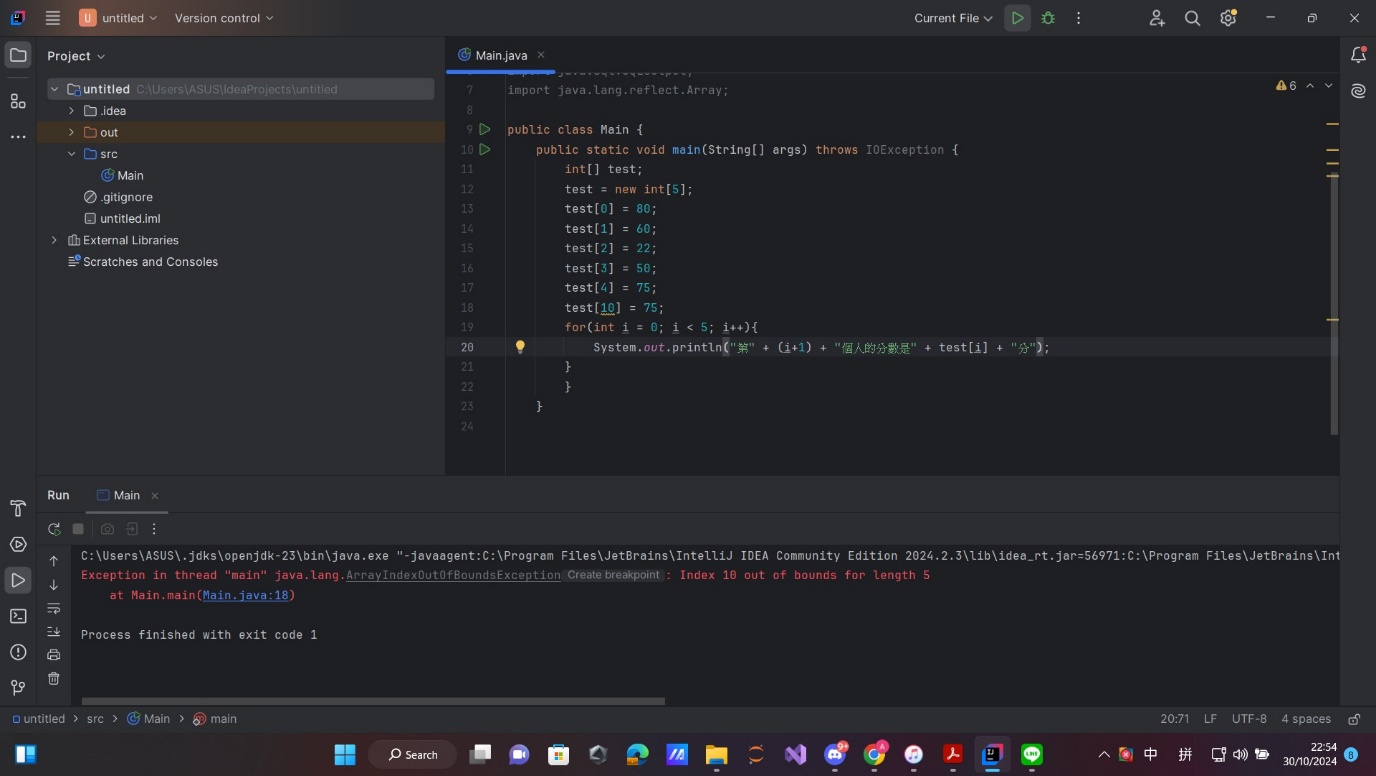


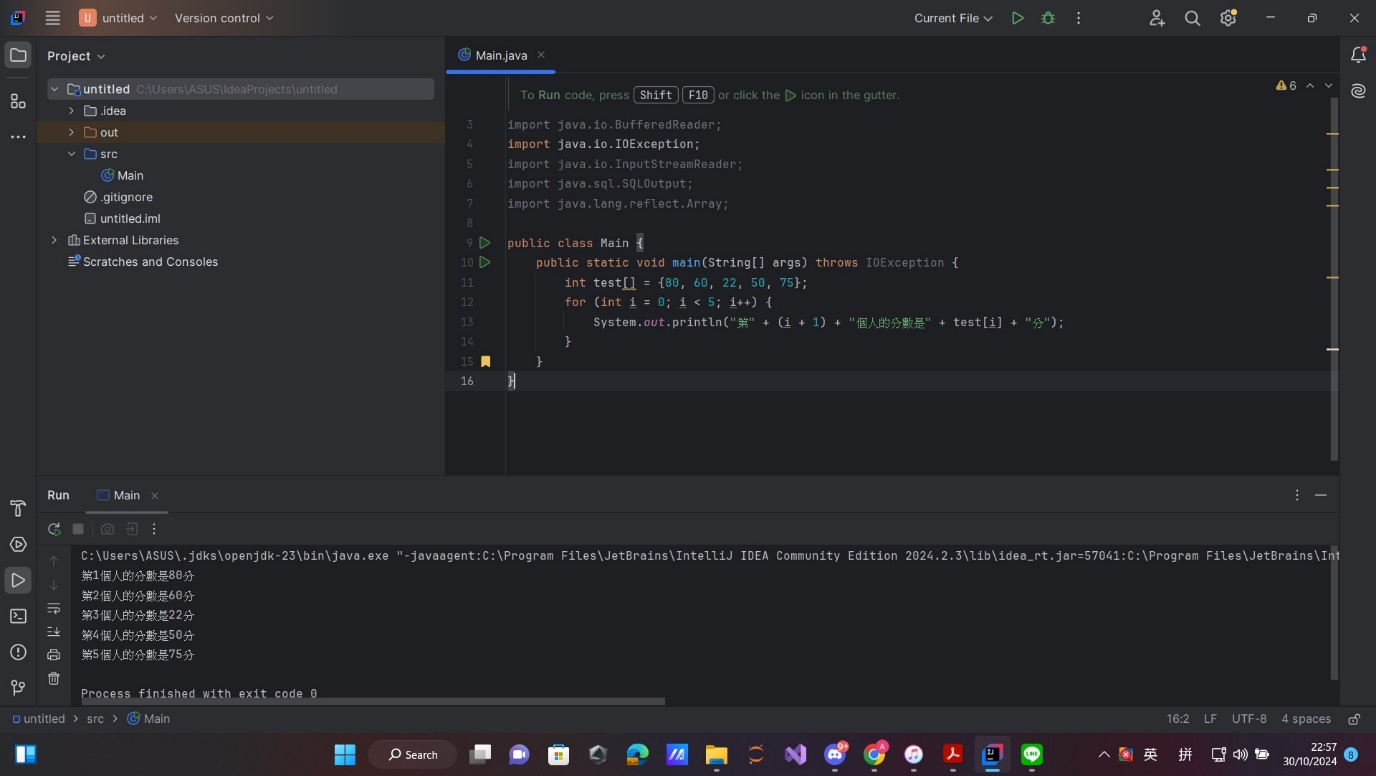


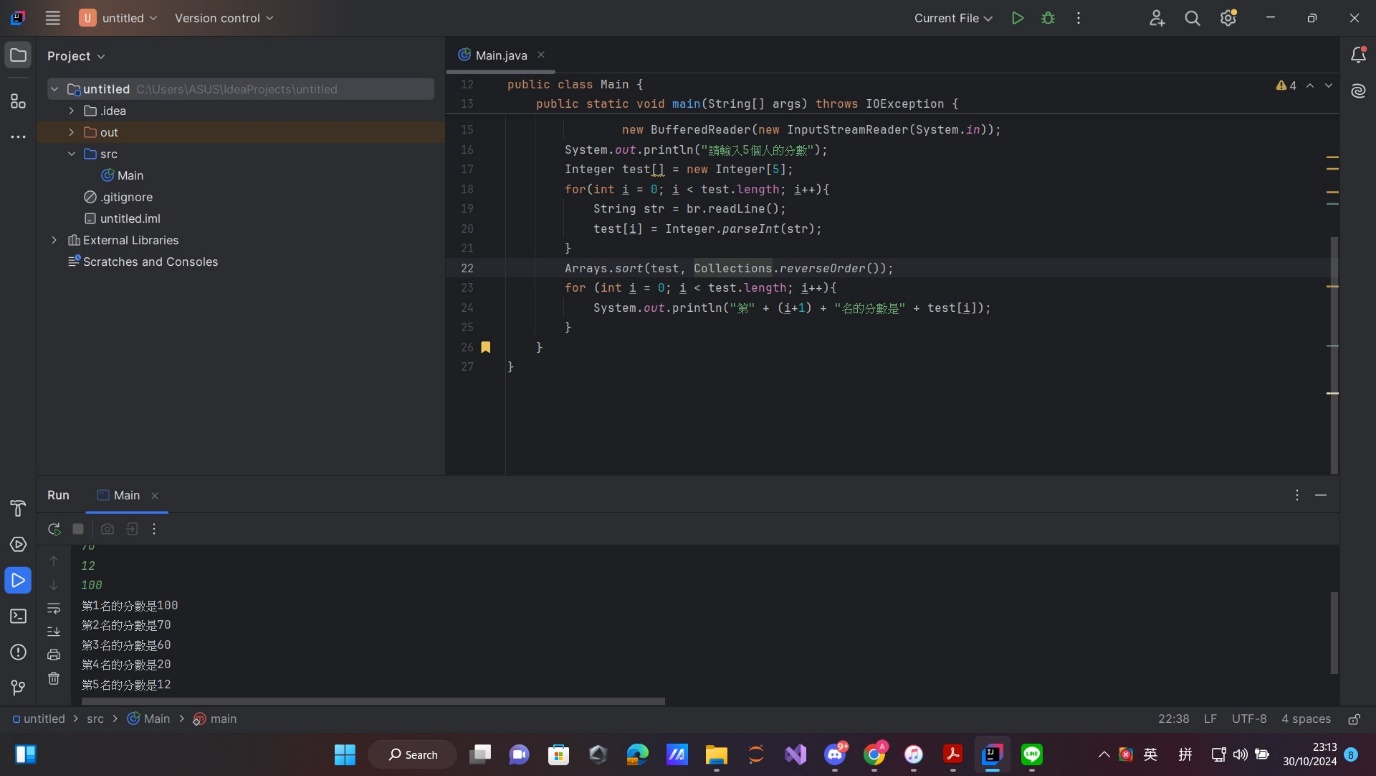


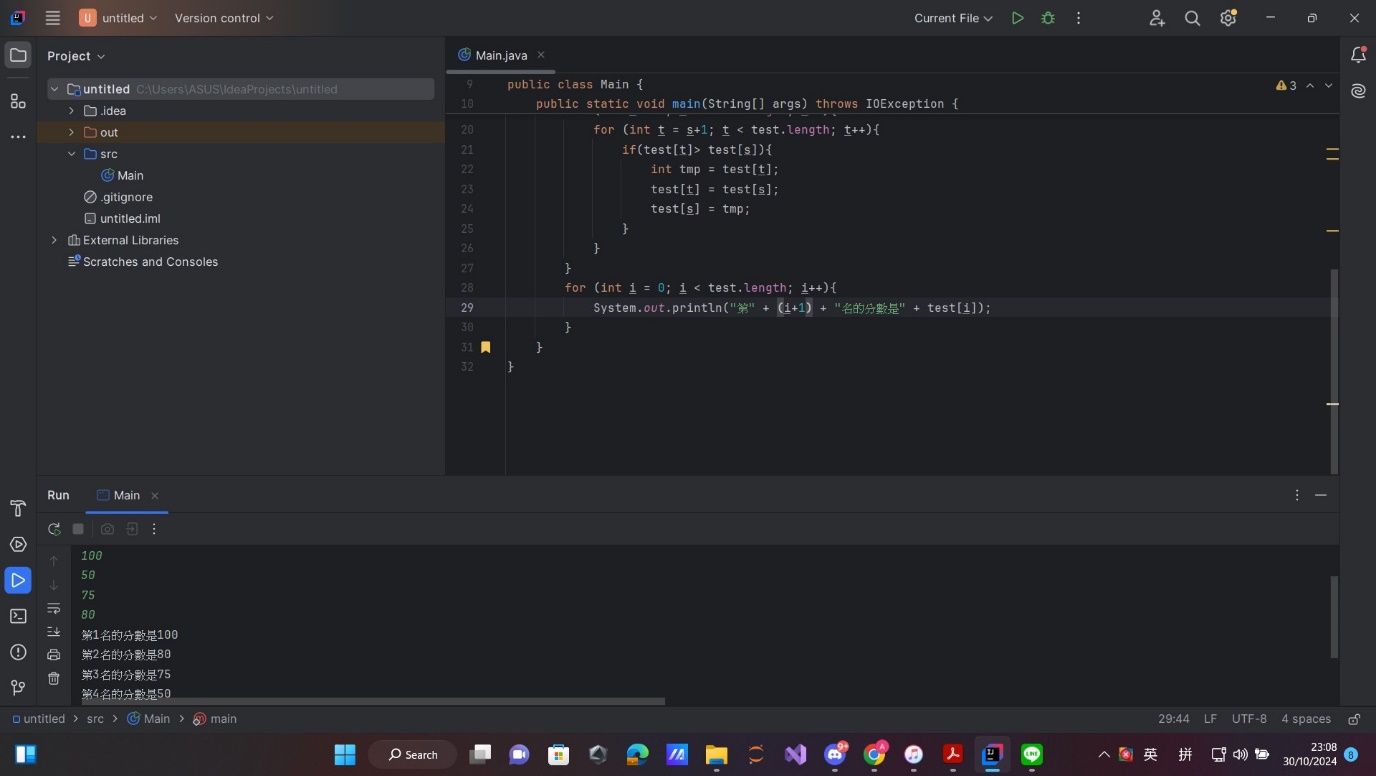


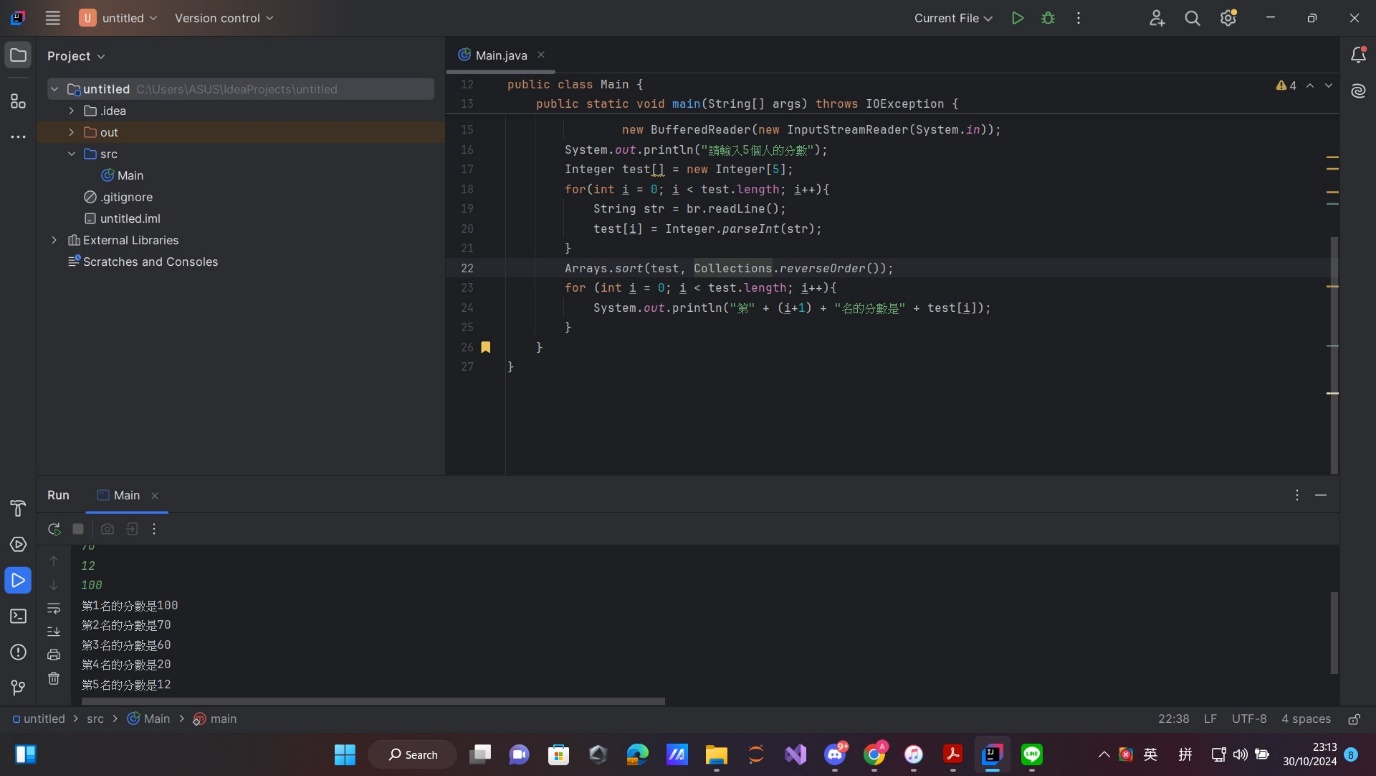


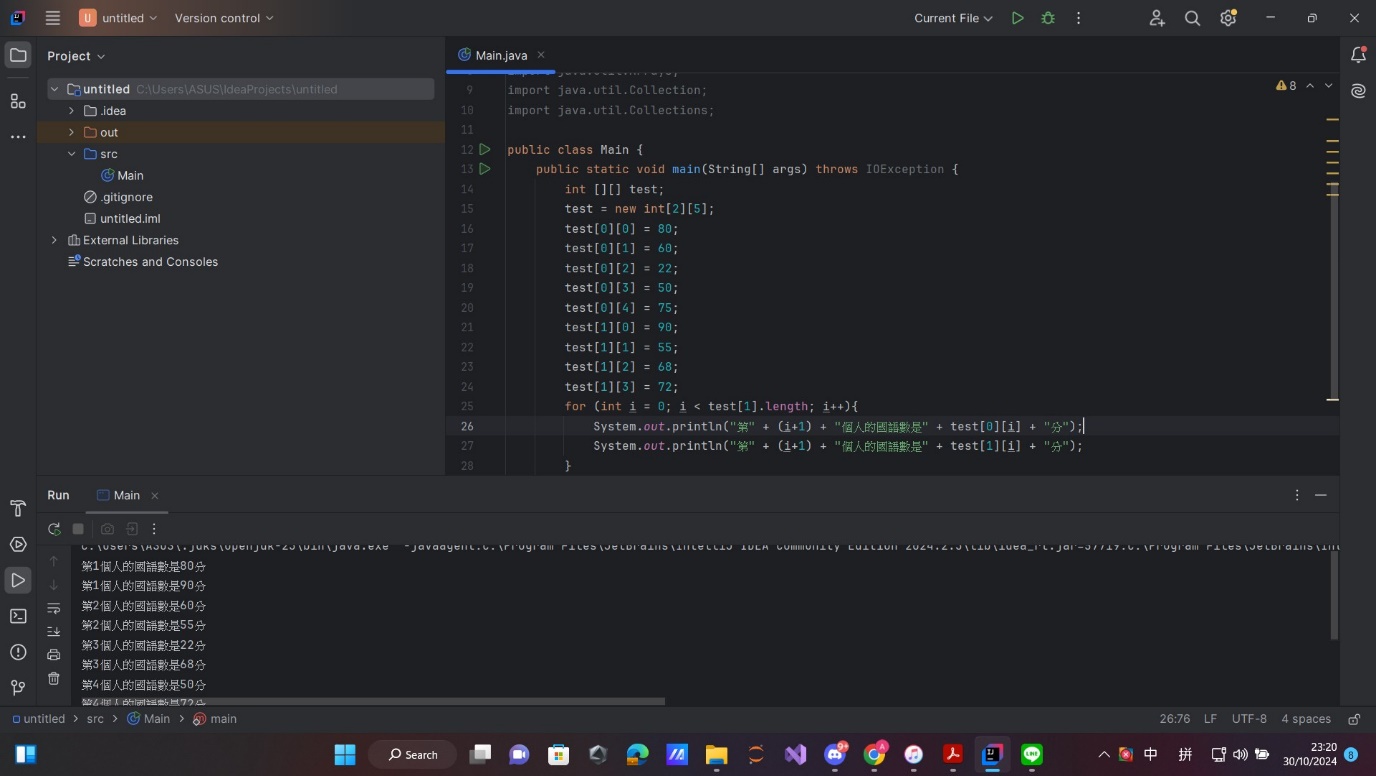












Question:

1. How do we use Java's BufferedReader to handle user input?
2. When should you choose between using if-else and switch-case?
3. What are some common uses of arrays in Java?

Chat GPT:

1. In Java, BufferedReader is used to read text from an input source (e.g., the console or a file), and it’s often used when handling user input in console applications. To set it up, wrap BufferedReader around an InputStreamReader that takes System.in as a parameter.
2. Choose if-else for conditions that are complex or don’t relate to a single variable. if-else can evaluate multiple variables, complex expressions, and ranges, which makes it versatile.

Use switch-case when you need to check a single variable against multiple possible constant values, such as a set of fixed options (like a menu). switch-case is often preferred for such scenarios because it makes code cleaner and can be faster for execution.

1. Arrays are commonly used in Java for tasks that involve storing lists of items of the same type, like student grades, product prices, or character sequences. Arrays are useful because they allow indexed access and fixed-size storage. Examples of array usage include:

**-. Storing sequences** of items that don’t change in size, such as days of the week.

**-. Sorting data** using array-sorting algorithms, especially when dealing with numerical data.

**-. Implementing matrix structures**, where multi-dimensional arrays (e.g., int[][]) can represent grids or tables.

In conclusion, this assignment introduced us to the fundamentals of Java programming, covering key concepts like BufferedReader for handling input, conditional statements (if, if-else, switch-case) for decision-making, and while loops with break for controlled repetition. We also worked with arrays to efficiently store and manage data. These basics provided a strong foundation, preparing us for more advanced programming concepts in the future.