Hello. My name is 이한얼 and my partner is 우정혁. Today I will be presenting the process of compiling and running our project code. Before started, we satisfied all the check lists. The key steps are as follows. Opening the code, compiling it, running the executable, and finally comparing the results with the original source code.

First, let's start by opening the code. In our case, we made the source code using Visual Studio Code. As you can see on the screen, the code is now ready to be compiled. So let me explain why this code was created. We used recursion instead of while of for loops to simplify it. And omitted the return type such as int or void to keep it shorter. By using logical operators and recursion, we reduced the code to a single line. Starting from 1, printf prints each number followed by a comma, with recursive calls continuing until i reaches 10000. This approach met the size limit while ensuring the program still functions as intended.

Next, we move on to compiling the code. We used the GCC compiler for this and the command used was as follows. During the compilation, a warning occurred because we did not include the <stdio.h> header, which is necessary for printf function. However, because our primary goal was to keep the code under 53 bytes, we ignored this warning as the compilation was still completed successfully.

Now that the code is compiled, we can run the executable. To do this, we used the command ./count2 in the terminal. As you can see on the screen, the program starts running, and it displays the expected output. This step confirms that our code has been successfully compiled and is functioning as intended.

To compare the results of both the original and new code, we run both executables and capture their output using the following command. This command redirects the output of count1 and count2 to two separate files, c1 and c2. The output of each program is now stored in these files for comparison. Finally, we compare the outputs of the original and new codes using the diff command. This command compares the contents of the two files, c1 and c2. If there are no differences, it will not return any output, which means that the results of both programs are identical. In our case, there were no differences.

This is the end of our presentation. Thank you for listening.