Group 07
Aspa, Angelica
Bober, Xyzriel Khrystien
Dela Rosa, Mico Von
Romero, Julez Deyn
Vitto, Diosh

Document Report

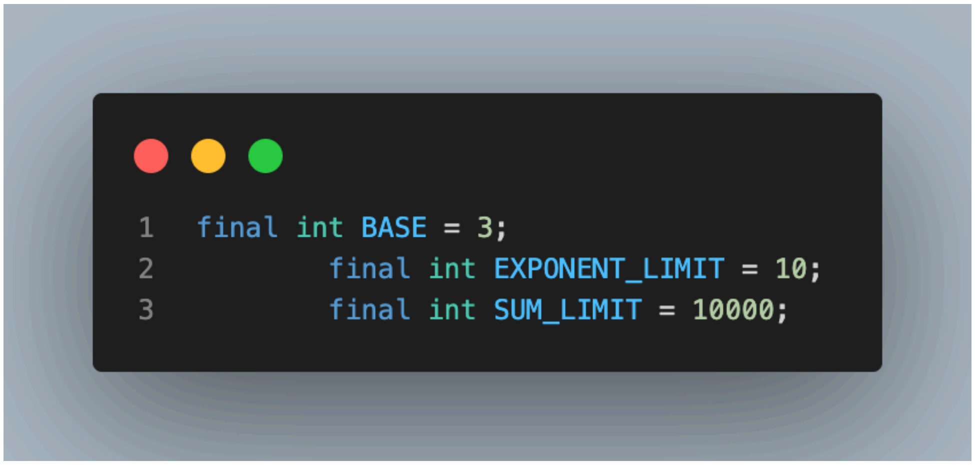
Problem Description

The programming problem provided is to generate a simple power table for a constant base with a certain exponent limit. The given program should calculate and print each power starting from base^1 until it reaches the set exponent limit. Simultaneously, it should compute the sum of all powers. If adding the next power causes the sum to exceed 10,000, it should immediately stop the program and display an error message.

Algorithm

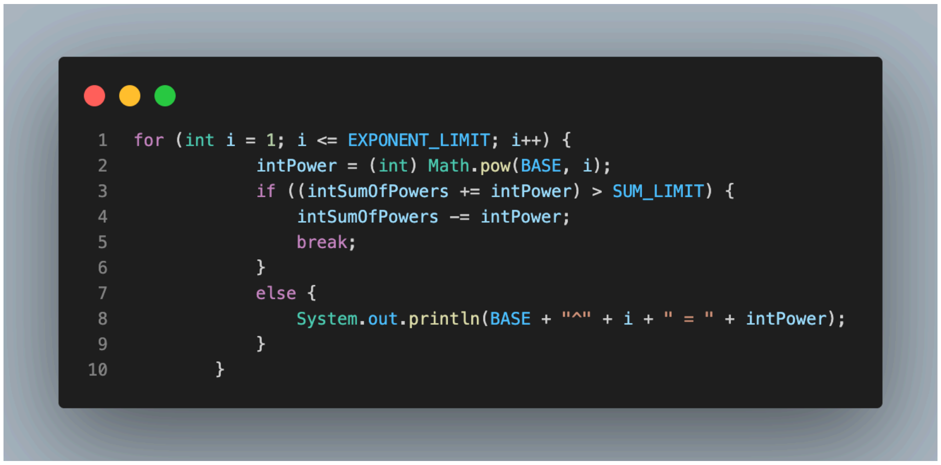
```
// Hardcode base and exponent limit
set BASE = 3
set EXPONENT_LIMIT = 10
set SUM_LIMIT = 10000
set intSumOfPowers = 0

// Loop each exponent starting from 1 until the limit
for exponent from 1 until EXPONENT_LIMIT do
    intPower = BASE ^ exponent
    intSumOfPowers = intSumOfPowers + intPower // Adds the power to sum

// Detection for exceeding 10000
if intSumOfPower > SUM_LIMIT
    intSumOfPower = intSumOfPower - intPower // Removes the last power that makes it exceed limit
    break
else
    print BASE ^ exponent = intPower // Prints the table
end if
end for

print “Sum of powers: “ + intSumOfPowers
print “The succeeding power makes the sum of powers exceed 10,000.” // Error handling message
```

Code Snippets



```
final int BASE = 3;
            final int EXPONENT_LIMIT = 10;
            final int SUM_LIMIT = 10000;
```

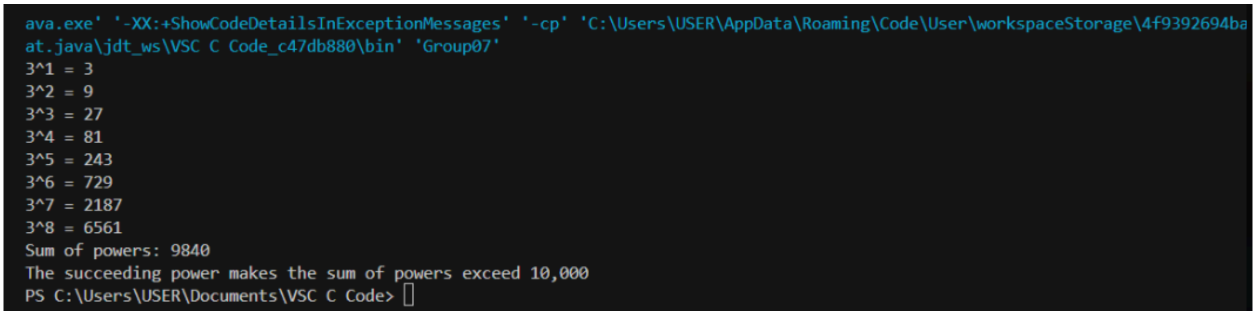
Figure 1. *Hardcoded base and exponent limit including the sum limit.*



```
for (int i = 1; i <= EXPONENT_LIMIT; i++) {
            intPower = (int) Math.pow(BASE, i);
            if ((intSumOfPowers += intPower) > SUM_LIMIT) {
                intSumOfPowers -= intPower;
                break;
            }
            else {
                System.out.println(BASE + "^" + i + " = " + intPower);
            }
        }
```

Figure 2. *Conditional statements that generate the power table (for loop) and checks if sum exceeds limit (if-else).*

Sample Output



```
ava.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\4f9392694ba
at.java\jdt_ws\VSC C Code_c47db880\bin' 'Group07'
3^1 = 3
3^2 = 9
3^3 = 27
3^4 = 81
3^5 = 243
3^6 = 729
3^7 = 2187
3^8 = 6561
Sum of powers: 9840
The succeeding power makes the sum of powers exceed 10,000
PS C:\Users\USER\Documents\VSC C Code>
```

Figure 3. *Sample output with error handling message.*

Challenges Encountered

There were only a few instances where the group had encountered problems during this activity. The programmers assigned were slightly confused by the instructions about hardcoding a base and exponent limit—whether or not it should be a user input or a constant but was later clarified right away. Additionally, due to the slight syntax difference between C and Java, it was somewhat tough to adjust to the language. Another instance was the adverse weather condition, making it slightly difficult for the documentors to start and finish early due to the intermittent internet connection.

Learning Outcomes

Overall, the group has learned to implement the proper naming conventions for identifiers and also create the proper and descriptive names that make the code easier to read and understand. Additionally, the group has learned the meaning of “hardcoding” variables, which would make it simpler for the group to understand the instructions for other programming activities. Lastly, through this activity, the group has learned to work together with balance and hopes that it continues for the entire semester.