

252-0027-00L: Introduction to Programming

Exercise Sheet 4

Submission: October 13, 2026, 23:59

Check out the new template with IntelliJ by selecting *Main Menu* → *Git* → *Pull...*, as in Exercise 2. If you run into problems, the [IntelliJ guide](#) will help you. Particularly helpful is the step described there of clicking the elephant icon on the right edge of IntelliJ and then pressing the *Reload All Gradle Projects* button (two circular arrows).

Note that in this exercise you will have several independent programs in the same IntelliJ project. Therefore, before you start a program, make sure that you have selected the correct file or opened it in the editor. *Do not forget to comment your program code!*

Task 1: Binary Representation

Write a program “Binaer.java” that reads in a positive number Z and then prints the [binary representation](#).

Hint: First find the largest number k such that the power of two $K = 2^k$ is less than or equal to Z . This is the first 1-bit; after that, you can subtract the power of two and go downward through the powers of two. If $K = 2^{k-1}$ is less than or equal to the remaining number, then the next bit is 1, otherwise 0, etc.

Example: For $Z = 14$, your program should output 1110.

Note that we do not use two’s complement here and only positive numbers should be represented.



[xkcd: Su Doku](#) by Randall Munroe
(CC BY-NC 2.5)

Task 2: Greatest Common Divisor

Write a program “GGT.java” that computes the greatest common divisor (GCD) of two positive integers using the Euclidean algorithm. This is an iterative computation based on the following observation:

1. If x is greater than or equal to y and x can be divided by y without remainder, then the GCD of x and y is equal to y ;

- otherwise (if x is smaller than y or x is not divisible by y), the GCD of x and y is the same as the GCD of y and $x \% y$.

First practice this algorithm by hand on a few examples and then write the Java program.

Example: For $x = 36$ and $y = 44$:

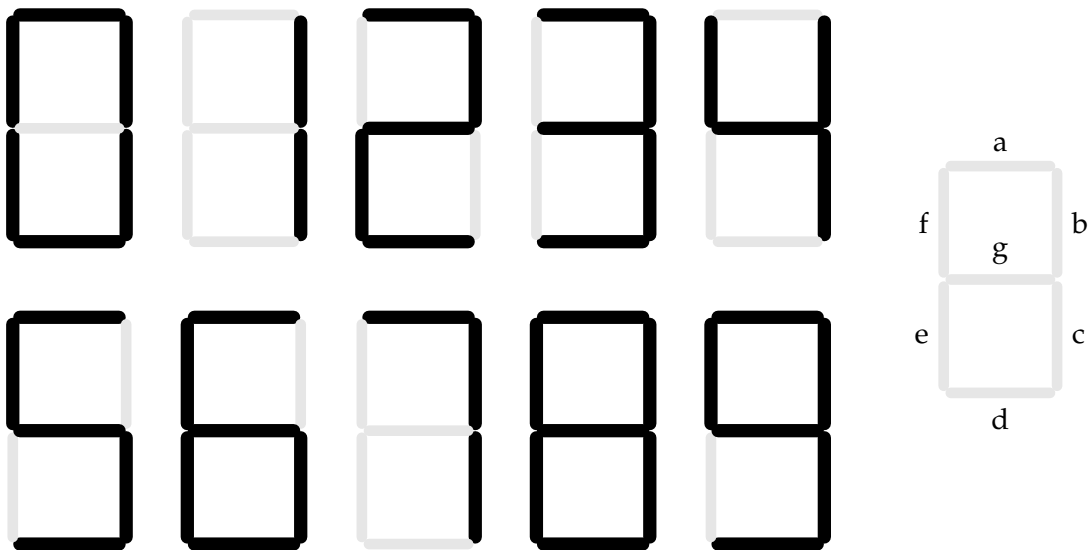
$$\text{GCD}(\underbrace{36}_x, \underbrace{44}_y) \stackrel{2}{=} \text{GCD}(44, \underbrace{36 \% 44}_{36}) \stackrel{2}{=} \text{GCD}(36, \underbrace{44 \% 36}_8) \stackrel{2}{=} \text{GCD}(8, \underbrace{36 \% 8}_4) \stackrel{1}{=} 4$$

Task 3: Number Recognition

For this task, we use a string to encode the lit segments of a [seven-segment display](#). The segments are labeled from a to g, as shown in the image. The encoding of a possible display is a string in which the letter x occurs exactly when the x-th segment of the display is lit. For example, the number 2 is encoded by 'abged'. For simplicity, you may assume that no letter occurs more than once in the encoding and that only the numbers 0 to 9 are encoded.

Write a program "Zahlen.java" that reads in a string encoding a display and prints the encoded number as an integer. Think about how many if blocks are needed to recognize every number.

Tip: You can use `str.contains("a")` to check whether a string `str` contains the letter 'a'.



Task 4: Calculations

Write your solution for this task in the class "Calculations.java". Tests for both methods are available in the IntelliJ project.

- Implement the method `Calculations.magic7(int a, int b)`. The method returns a boolean. The method should return true if at least one of the following conditions is

fulfilled: the first parameter is 7, the second parameter is 7, the sum or one of the differences of the parameters is 7. Otherwise, the method should return `false`. The differences of the parameters `a` and `b` are defined as `a - b` and `b - a`.

Examples

- `magic7(2,5)` returns `true`.
- `magic7(7,9)` returns `true`.
- `magic7(14,7)` returns `true`.
- `magic7(7,14)` returns `true`.
- `magic7(5,6)` returns `false`.

Hint: With the function `Math.abs(num)`, you can obtain the absolute value of a number `num`.

2. Implement the method `Calculations.fast12(int z)`. The argument `z` is nonnegative. The method returns a boolean. The method should return `true` if `z` is close to a multiple of 12. A number `x` is close to a number `y` if one of the numbers is at most 2 greater or smaller than the other number. Otherwise, the method should return `false`.

Examples

- `fast12(12)` returns `true`.
- `fast12(14)` returns `true`.
- `fast12(10)` returns `true`.
- `fast12(15)` returns `false`.

Task 5: Scrabble

In this task, you should lay Scrabble tiles using ASCII art on the console. Complete the method `drawNameSquare` in the class `Scrabble`. This method takes a name as a string parameter and should print the name as Scrabble tiles arranged in a square on the console (`System.out`). If, for example, the string `Alfred` is passed, the following image should be output:

```
+---+---+---+---+---+---+
| A | L | F | R | E | D |
+---+---+---+---+---+---+
| L |           | E |
+---+           +---+
| F |           | R |
+---+           +---+
| R |           | F |
+---+           +---+
| E |           | L |
+---+---+---+---+---+---+
| D | E | R | F | L | A |
+---+---+---+---+---+---+
```

Your code only needs to support names of length 3 or longer. For a name of length 3, e.g. `Jim`, the output should look like this:

```
+---+---+---+
| J | I | M |
+---+---+---+
| I |   | I |
+---+---+---+
| M | I | J |
+---+---+---+
```

Note that your program must not produce any output other than the Scrabble square. Use `System.out.print(...)` when no automatic line break is required. Use `System.out.println(...)` when a new line should begin afterwards. In your loops, also make sure to output the first and last required positions exactly once each.

Task 6: Operators: Fundamentals

Note: This task is completed as a [Moodle quiz](#).

Task 7: Operators: Advanced

Note: This task is completed as a [Moodle quiz](#).