Problem 1

Write a program which can be used as a (very) simple calculator. The program reads, using a **Scanner**, two numbers (of type **double**) and, as a **String**, the symbol of an arithmetic operation (+ , − , ∗ or /). Then it displays the result (sum, difference, product or ratio of the two numbers). In order to compare strings, say `s1` and `s2`, use function **equals** (`s1.equals(s2)` returns **true** or **false**).

Problem 2

Write a program reading integers until zero is entered and printing the length of the longest sequence of consecutive numbers of the same value (and this value).
For example, for

```
2 2 2 2 3 3 3 2 6 6 6 0
```

the result should be

```
Longest sequence: 4 times 2
```

for

```
2 2 2 3 3 3 3 2 6 6 6 0
```

the result should be

```
Longest sequence: 5 times 3
```

and for

```
2 2 3 3 2 3 6 6 6 0
```

it should be

```
Longest sequence: 3 times 6
```

Do not use arrays, strings or any other kind of collections.

Problem 3

Write a program which reads four integer numbers and prints the difference between the largest and the smallest of them. Don't use arrays, strings or collections.

Problem 4

Write a program which reads an odd natural number n and writes on the console a symmetric 'kite' consisting of asterisks: in the first line n stars, in the second line $n-2$ stars, ... , in n-th – one star. For example, for $n = 7$ the output should look like this:

```
*******
 *****
  ***
   *
```

Problem 5

Write a program which reads two natural numbers, w and h, and then prints a rectangle with width w and height h. For example, if the numbers are 7 and 5, the result should look like this:

```
*******
*     *
*     *
*     *
*******
```

Problem 6

Write a program which finds the greatest common divisor (GCD) of two numbers read from the user. Use Euclid's algorithm (*Elements,* Book VII).

In order to find the GCD of number a i b ($a, b \in \mathbb{N}$) repeat the sequence:

1. if $a = b$, then GCD=a and END;
2. if $a > b$, then decrease a by b or, if $b > a$, then decrease b by a;

Note that the algorithm requires only testing equality and subtracting. For those mathematically inclined: show that the algorithm will always stop, i.e., that after a finite number of iterations the condition in step 1 will be satisfied (and both a and b will then be equal to GCD of the two numbers that we started with).

Problem 7

Write a program printing the multiplication table for numbers from 1 to n. For example, if n is 12, the table should look like this:

```
 1   2   3   4   5   6   7   8   9  10  11  12
 2   4   6   8  10  12  14  16  18  20  22  24
 3   6   9  12  15  18  21  24  27  30  33  36
 4   8  12  16  20  24  28  32  36  40  44  48
 5  10  15  20  25  30  35  40  45  50  55  60
 6  12  18  24  30  36  42  48  54  60  66  72
 7  14  21  28  35  42  49  56  63  70  77  84
 8  16  24  32  40  48  56  64  72  80  88  96
 9  18  27  36  45  54  63  72  81  90  99 108
10  20  30  40  50  60  70  80  90 100 110 120
11  22  33  44  55  66  77  88  99 110 121 132
12  24  36  48  60  72  84  96 108 120 132 144
```

In order to print a number on four places, padding it from the left with spaces for shorter numbers, you can use `System.out.printf("%4d",n)`.
