

CS1 — Homework Module 6: Recursion

October, 2022

Exercise 1

The factorial function (represented by the exclamation mark, $!$) is a mathematical formula to multiply all whole numbers from a chosen one down to 1. This operation is very common in probability when calculating combinations and permutations.

Examples:

$$3! = 3 \times 2 \times 1 = 6$$

$$7! = 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 5040$$

$$1! = 1$$

Implement a method that, given an integer as the input parameter, calculates the factorial of this number.

Important: *debugging this method and analysing what is happening step-by-step may be very helpful to understand the concept of recursion.*

Exercise 2

According to the Cambridge dictionary, a palindrome is *a word or group of words that is the same when you read it forwards from the beginning or backwards from the end*. Some examples could be:

- civic
- mom
- radar
- rotavator
- eva, can I see bees in a cave?
- never odd or even
- (Spanish) Sé verlas al revés
- (Valencian) Tip, el pastor ara farà rots a ple pit

Implement a piece of code that returns whether a word given as an input parameter is a palindrome or not and returns a Boolean value. Then, in the `main` method, print a message depending on the value returned.

Exercise 3

In mathematics, Pascal's triangle is a triangular array of the binomial coefficients that arises in probability theory, combinatorics, and algebra (Wikipedia). You can learn more about it [here](#). Write a method which must receive an integer as an input parameter and print the first x rows of

this triangle on the screen, being x the value of the input parameter. An example for $x=10$ would be:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

Exercise 4

It is known in mathematics that if you take any natural number n (i.e. 1, 2, 3), apply the following rules:

1. if n is even: divide n by 2
2. if n is odd: multiply by 3 and add 1

and continue to do this with the new number you obtain, you will at some point always reach the number 1.

Implement a Java Program that simulates this process, i.e. that reads in an integer number and then uses the rules listed above to build a sequence of numbers until it reaches the value 1. Your program should print out all the numbers that make up the sequence together with, in the end, the number of steps it took to reach the value 1.

Last time you were required to implement this with a while loop. This time, of course, you should build a recursive solution.

Exercise 5

Implement some of the “Simple Array Algorithms” we covered in class recursively. The algorithms to implement are:

- Finding the maximum/minimum
- Finding a value

*Also the other algorithms can be implemented recursively. For example, implementing a recursive version of the **Binary search algorithm** is a great exercise for the exam! In fact, it would constitute a big step towards passing this module if you manage to implement all exercises on iteration from previous sessions recursively and all exercises from this session also iteratively. Be aware though that you should not consider this easy.*