02393 C++ Programming Exercises

Assignment 11

To be handed in via Autolab — `https://autolab.compute.dtu.dk/courses/02393-E23/assessments`

1 Fibonacci trees

Write a program that, given a non-negative integer n, provides some information about the computation of the Fibonacci function $F(n)$ recursively defined as follows:

$$F(0) = 1 \qquad F(1) = 1 \qquad F(n) = F(n-2) + F(n-1)$$

For example, if the input is 3, a recursive implementation of $F(n)$ (according to the definition above) would result in the following tree of recursive calls:

```
    F(3)
   /    \
  F(2)  F(1)
 /    \
F(1)  F(0)
```

The program should output:

- all Fibonacci numbers computed, shown by traversing the above tree in *pre-order* (see below);
- the size of the tree;
- the tree depth (a.k.a. height);
- the number of leaves (i.e. nodes without sub-trees, which in this case correspond to the base cases of the recursive formulation).

So, if the input is 3, the format of the output should look like this:

```
Call tree in pre-order: 3 2 1 1 1
Call tree size: 5
Call tree depth: 3
Call tree leafs: 3
```

Pre-order traversal of the Fibonacci tree. Consider the tree of recursive calls above. If we replace the calls $F(i)$ by their respective results, the tree would look like this:

```
    3
   / \
  2   1
 / \
1   1
```

And the pre-order traversal of such a tree would be `3 2 1 1 1`, where:

- the first `1` refers to the left son of `2`;
- the second `1` refers to the right son of `2`;
- the third `1` refers to the right son of `3`.

In other words, for each node:

- we first print the value of the node;
- then we descend in the left sub-tree (if it exists);
- and then we descend in the right sub-tree (if it exists).

Hint. Use a tree structure to build the tree of recursive calls, and then output the required information by traversing the tree structure itself.