Exercise #2	Algorithms and Data Structures	
	Topic: Recursion	Version: 1.0 / 2019
	Prepared by: dr inż. Grzegorz Łukawski & dr inż. Barbara Łukawska	

1) Recursion

1.1) The definition

Recursive object – an object, partially consisting of itself, or which definition refers to itself. The recursion allows for defining an infinite set of objects using a finite expression.

Characteristics of recursive algorithms:

- the ending is clearly defined;
- main problem is divided into elementary problems, which solutions are known.

1.2) Example

Computing a factorial of "n" using recursion:

$$0! = 1; \quad if\ n > 0 \quad then \quad n! = n * (n - 1)!$$

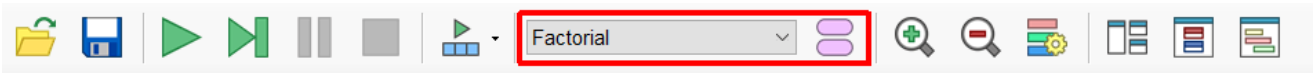
Implementation in C++:

```
#include <iostream>
using namespace std;

int factorial(int n) {
	if (n == 0) return 1;
	else { return(n*factorial(n - 1)); }
}
int main() {
	int num;
	cin >> num;
	cout << factorial(num) << endl;
	return 0;
}
```

1.3) Recursion in Flowgorithm

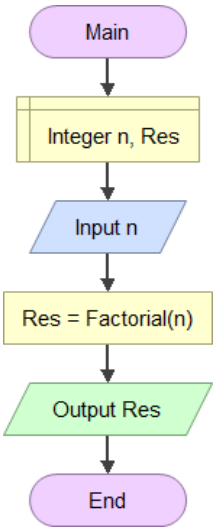
For building a recursive algorithm, a function must be defined. To create a new function, use the menu option "Program -> Add Function" or use the corresponding icon on the toolbar (see below). Arguments and the returned value of a function should also be defined.

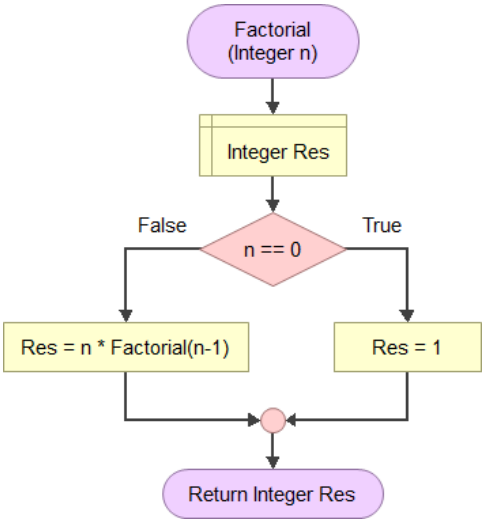


You can switch between the "Main" function and any other one using the choice button on the toolbar. To call a function, use its name and pass arguments in "Assign" box, similarly as in C (see example below).

1.4) Computing factorial in Flowgorithm

The recursive version of computing a factorial in Flowgorithm is presented below:





---

2) Exercises

Exercises may be implemented in C/C++ **or** Flowgorithm. Two solutions should be implemented for each exercise: **iterative and recursive** version. These versions should be compared (e.g. using debugger or by displaying variable values in the console).

A) Compute the result of $\boldsymbol{a^n}$ (*a* power *n*)
B) Check whether a word (entered as “string” or “char[]”) is a **palindrome**
C) Find *n*-th value of the **Fibonacci sequence**
D) Compute *n*-th value of the **geometric sequence**, for given a_0 (first value) and *q* (multiplier)