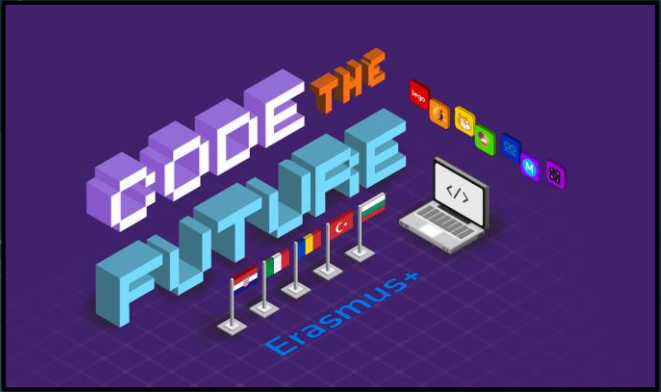
Giovanni Malafarina
Istituto Tecnico Tecnologico



Programming Language Algorithm

Prof. Claudio Cherubino

Code The Future
2019-1-TR01-KA229-074007
Soverato - Italy 9th-13th December 2019
1st Joint Staff Training

Co-funded by the
Erasmus+ Programme
of the European Union

Let me talk about a topic that will allow us, at the end of the presentation, to create an algorithm using a visual method with a program called "Flowgorithm".

So let's see how to represent algorithms without a classic programming language.

Summary

Before programming...

What an algorithm is

Algorithms in our lives

Pseudocoding

Flowcharts

Flowgorithm

Exercise

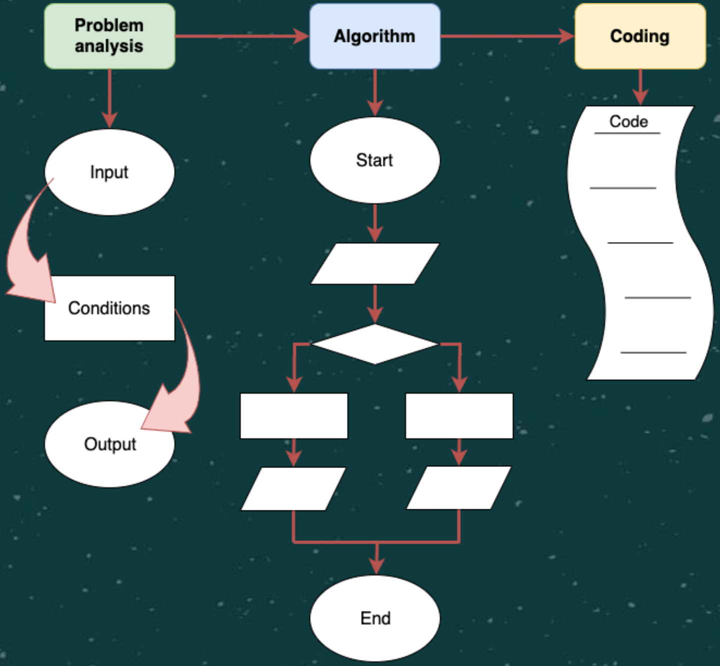
Before programming...

Solving a problem means obtaining a result starting from a certain initial state and following an appropriate procedure.

We need to analyze the problem by identifying the input data and, after processing, we shall obtain the output data.

Before writing the code to be executed we can "draw" a series of instructions that can solve the problem.

Then we can translate these instructions into any programming language.



What is an algorithm?

Every program we see is an algorithm! An algorithm is a list of rules to follow in order to solve a problem.

It has three fundamental **characteristics**:

- It must have a beginning and an end. A program cannot crash during the result creation process.
- Every single word of a program must refer precisely to an action to be performed, or to a value to be recovered.
- With the same input data our algorithm must reach the same result

Algorithms in our lives

"**Algorithm**" is a word that many know and that is often associated only with the world of computing. However, most of our daily life can be compared to an algorithm.

An example is the procedure displayed in any culinary recipe. So our mom's cookbook is a real book of algorithms!

Algorithms in our lives

Let's think of a traditional Italian recipe: Pizza.

Assuming that the dough is ready, proceed as follows:

1. roll out the dough and form a disk;
2. add the tomato;
3. add the mozzarella;
4. bake for 3 minutes;
5. serve it;
6. eat right away!!!

Pseudocoding

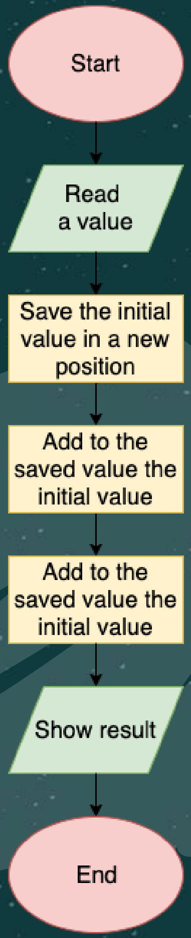
An algorithm cannot be interpreted directly by the computer. For this activity there are programming languages. These are "translators" who transform something comprehensible to humans into something understandable by our computer.

```
1. Start
2. Request the initial number
3. The result is the initial number
   multiplied by 3
4. Show the result
5. End
```

Pseudocoding

So what is pseudocoding?
It is a series of instructions, one per line.
Flowcharts are the most commonly used ones.

What we see is the flowchart of a number multiplied by 3


Start
Read a value
Save the initial value in a new position
Add to the saved value the initial value
Add to the saved value the initial value
Show result
End

Flowcharts

Flowcharts are diagrams that visually describe how the execution of a program proceeds. They are not related to a specific language.

It is a visual language understandable to all programmers.

The flow chart also helps the programmer to correctly describe an algorithm.

Flowcharts

Each type of instruction that can be inserted into a program has its own symbol and each of the three fundamental programming structures (**sequence**, **selection** and **iteration**) can be represented.

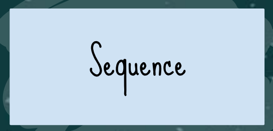
There are also special symbols (program start, program end) that do not represent actual instructions but they are useful for the construction of the flow chart.

Flowcharts

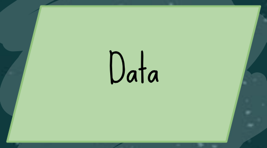
Main Symbols

Flowcharts - Main Symbols

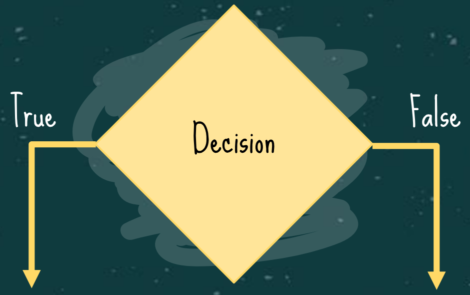
Sequence instructions



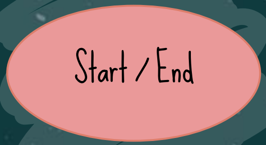
Input / output instructions



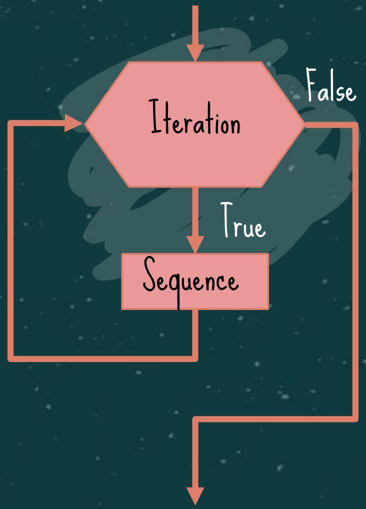
Selection instruction



Start / End flowchart



Iteration instruction (i.e.)

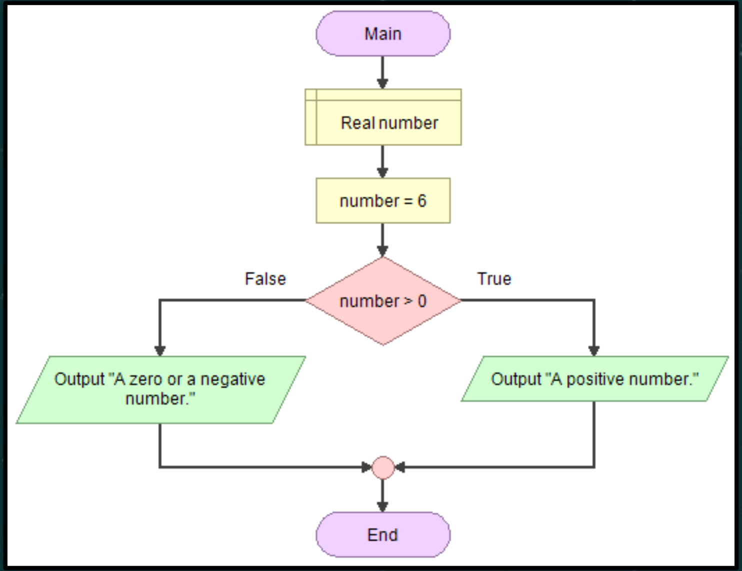


These are the representations of the basic instructions

Flowgorithm

We use **Flowgorithm** to create flowcharts.

Flowgorithm is a free beginner's programming language that is based on simple graphical flowcharts. By using flowcharts, we can concentrate on programming concepts. We can also run your programs directly in Flowgorithm.
Once we understand programming logic, it is easy for us to learn one of the major languages.
Flowgorithm supports multiple spoken languages.


Main
Real number
number = 6
False
number > 0
True
Output "A zero or a negative number."
Output "A positive number."
End

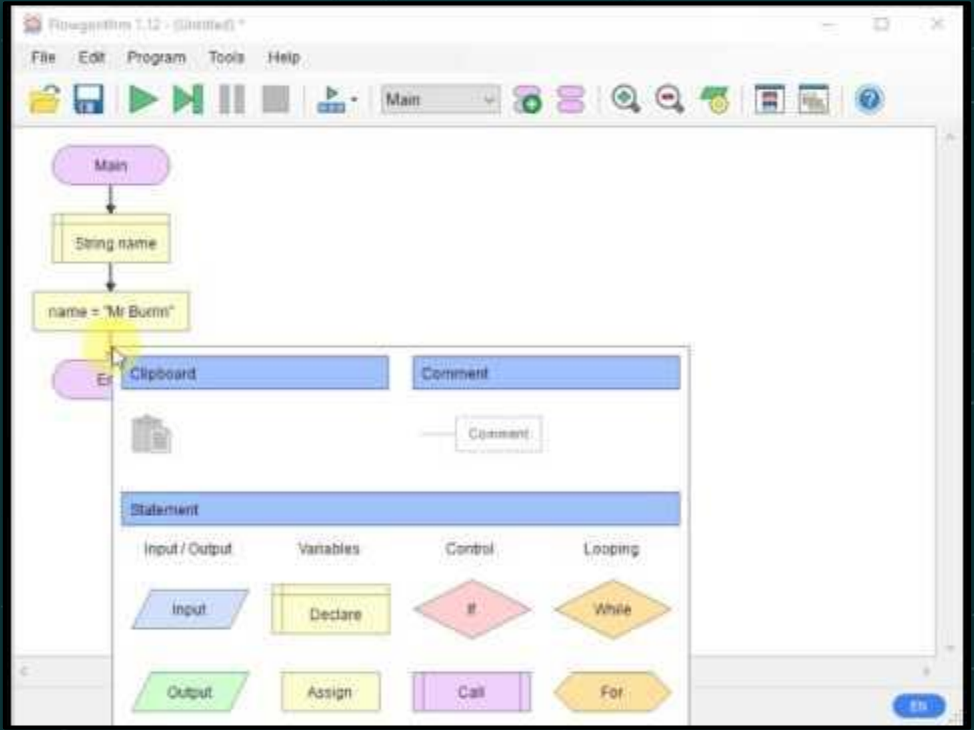
Downloading Flowgorithm

- http://www.flowgorithm.org/download/index.htm

- Portable Version:
  http://flowgorithm.altervista.org/

Flowgorithm - First Program

Following this short video we can create our first program with Flowgorithm



$\sqrt{123}$

Let's Try!

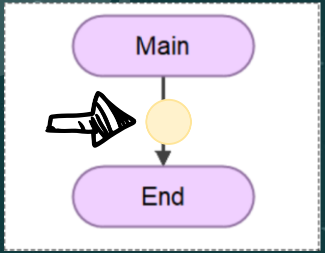
Let's try a little exercise together.
We multiply a number by 3.

STUDY
HARD!

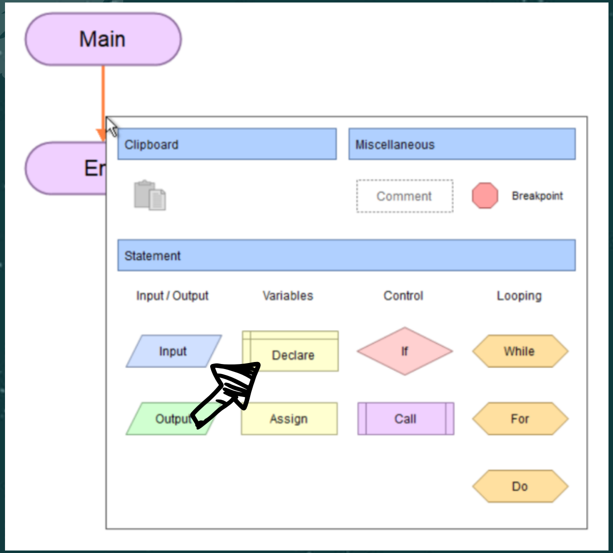
+×÷

Let's go, open Flowgorithm and follow the next steps !

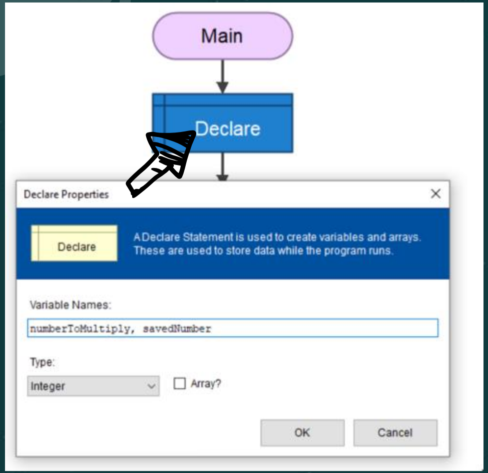
01



This is the starting condition. By clicking on the arrow we can add new instructions.



Now we have to declare all the variables necessary for our program. So, we select "Declare"



A Declare Statement is used to create variables and arrays. These are used to store data while the program runs.

Variable Names: numberToMultiply, savedNumber

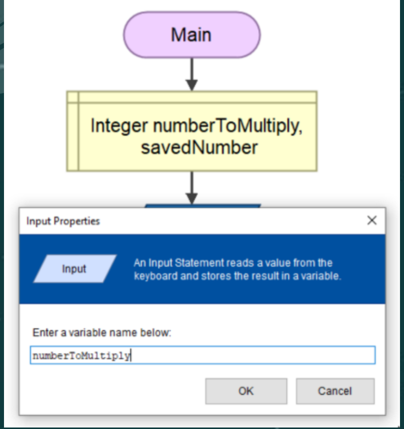
Type: Integer

☐ Array?

Now we double click on the "Declare" box. A window will appear where we can write our variables (numberToMultiply, savedNumber).

Next steps...

02



Main

Integer numberToMultiply, savedNumber

Input Properties

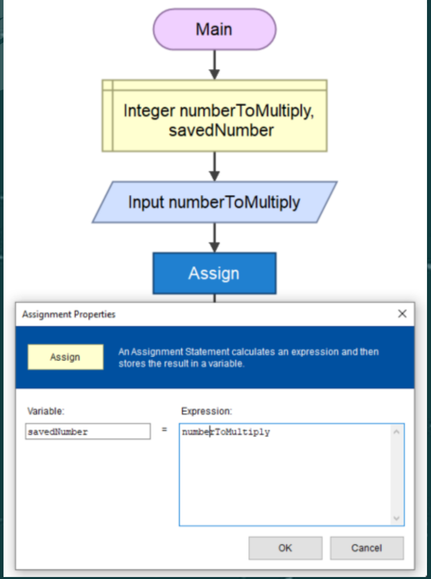
Input

An Input Statement reads a value from the keyboard and stores the result in a variable.

Enter a variable name below:

numberToMultiply

Let's read the value of the number to multiply. We insert an "Input instruction" by clicking on the arrow above the declaration box. We type the var name to read and press "OK"



Main

Integer numberToMultiply, savedNumber

Input numberToMultiply

Assign

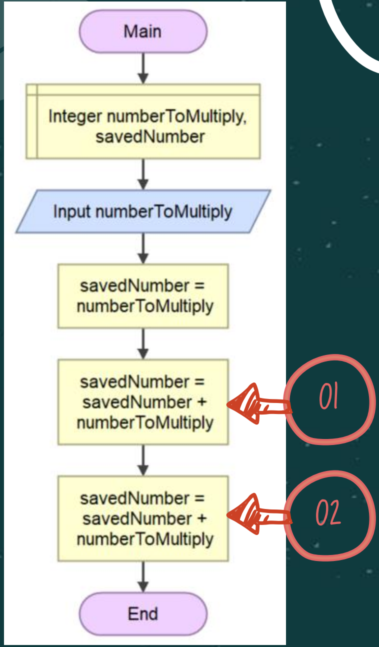
Assignment Properties

Assign

An Assignment Statement calculates an expression and then stores the result in a variable.

Variable: savedNumber = Expression: numberToMultiply

We save the initial number in the variable "savedNumber" through an "Assign Instruction".



Main

Integer numberToMultiply, savedNumber

Input numberToMultiply

savedNumber = numberToMultiply

savedNumber = savedNumber + numberToMultiply

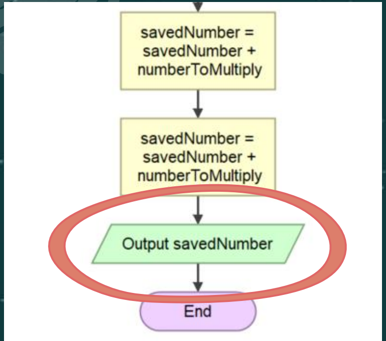
savedNumber = savedNumber + numberToMultiply

End

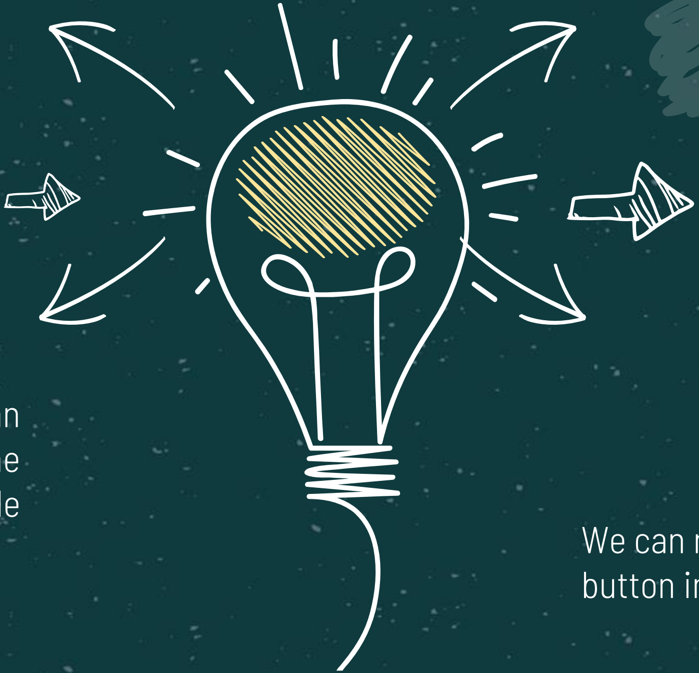
We insert 2 "Assign Instruction" to add "numberToMultiply" to "savedNumber" variable. By this way we have multiplied the initial number by 3.

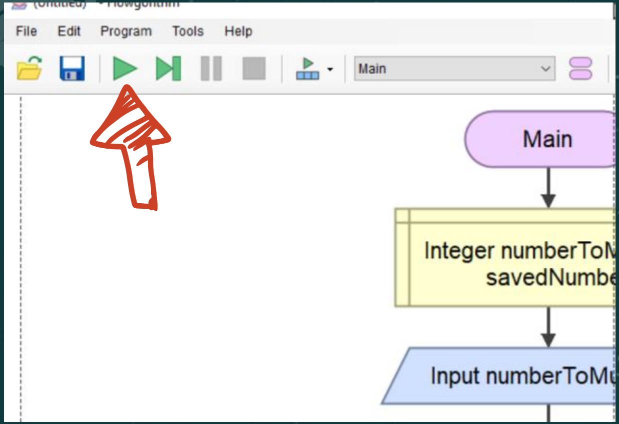
We're almost done

03


savedNumber = savedNumber + numberToMultiply
savedNumber = savedNumber + numberToMultiply
Output savedNumber
End

To display the result we add an output instruction, with the name of the variable "savedNumber"

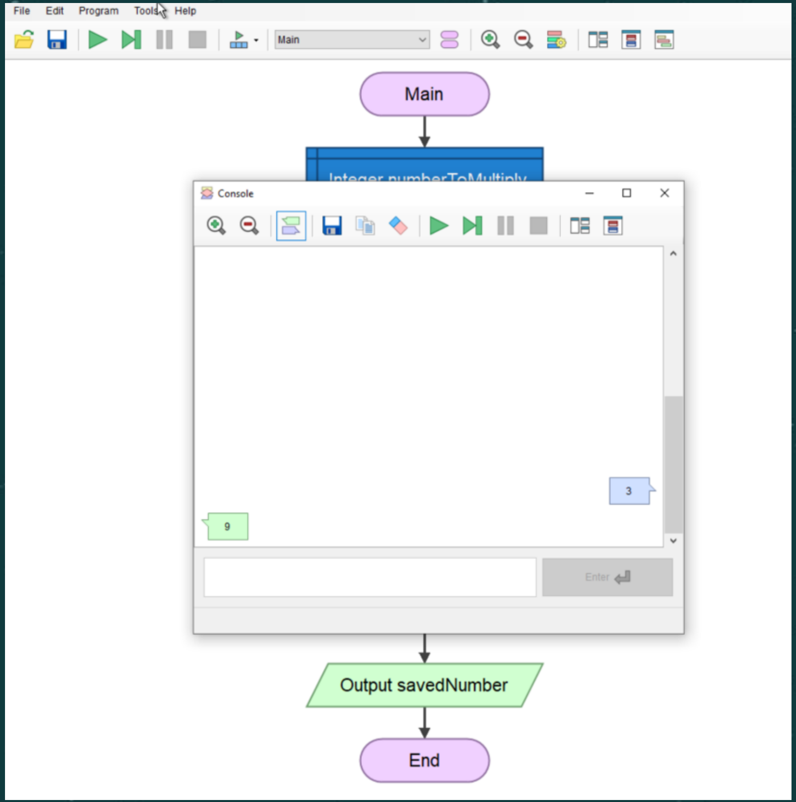




We can now run the algorithm by clicking on the "Play" button in Flowgoritm.

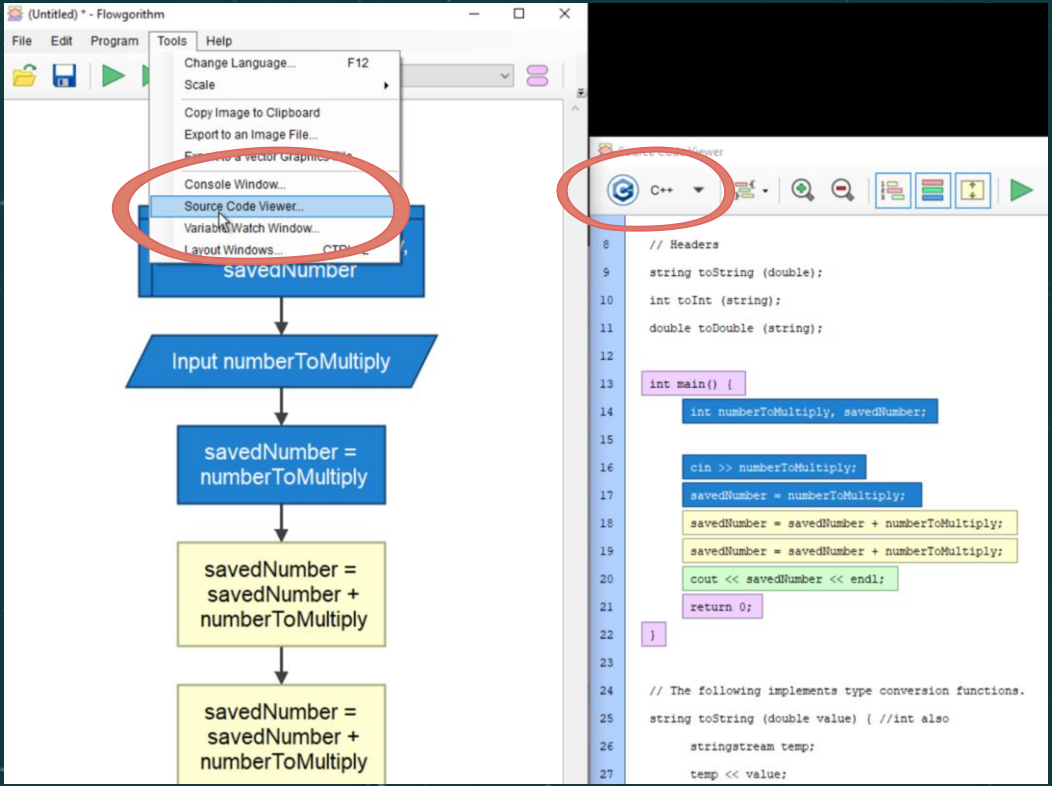
The result!

04



We will see the result of our algorithm in the Flowgorithm console. When requested we will enter the value of the initial number.

Source Code Viewer



```
// Headers
string toString (double);
int toInt (string);
double toDouble (string);

int main() {
    int numberToMultiply, savedNumber;

    cin >> numberToMultiply;
    savedNumber = numberToMultiply;
    savedNumber = savedNumber + numberToMultiply;
    savedNumber = savedNumber + numberToMultiply;
    cout << savedNumber << endl;
    return 0;
}

// The following implements type conversion functions.
string toString (double value) { //int also
    stringstream temp;
    temp << value;
```

With the "Source Code Viewer" tool it is possible to obtain the algorithm code in one of the main programming languages like C ++, PHP, Java.

EXERCISE

Implement the following algorithms with Flowgorithm:

- Read two numbers, make the sum and visualize the result.
- Sum the numbers from 1 to 10 and display the result.

Giovanni Malafarina
Istituto Tecnico Tecnologico

Claudio Cherubino

THANKS!

Co-funded by the
Erasmus+ Programme
of the European Union

"This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein."