# How do computers read code?

## I – Communication practice

**With your partners, prepare a short presentation (one or two slides) of one of the following topics.**

*The given links are only a starting point of the research that you will have to make to complete your knowledge on the topic.*

1. **The toolchain**
   a) What are the main components of a toolchain and, briefly, the role of each one? [W↗]
   b) What is the GCC toolchain? Describe it in particular for *C* and *C++* languages. [W↗]
   c) Are there other toolchains for *C* and *C++* languages? Describe one that is emblematic. [W↗]

2. **Preprocessing**
   a) In a toolchain, what is a preprocessor? Describe its purposes and results. [W↗]
   b) What is the preprocessor used for the *C* language family? Present it. [W↗]
   c) What are the principle and the interests of conditional compilation? Give an example. [W↗]

3. **Lexical analysis**
   a) In compiling, what is lexical analysis? Describe its purposes and results. [W↗]
   b) In C language, what are the main tokens that a lexer may find? Treat an example. [↗]
   c) Which lexer is used in a GCC compiler? Describe its main characteristics. [W↗] [↗]

4. **Syntactic analysis**
   a) In compiling, what is syntactic analysis? Describe its purposes and results. [W↗]
   b) What is an abstract syntax tree? Treat an example generated from a piece of *C* code. [W↗] [↗]
   c) Which parser is used in a GCC *C* compiler? Describe its main characteristics. [W↗] [↗]

5. **Semantic analysis**
   a) In compiling, what is semantic analysis? Describe its purposes and results. [W↗]
   b) What is a symbol table? Treat an example generated from a piece of *C* code. [W↗]
   c) What is type checking? What kind of type checking is performed by *C* compilers? [W↗]

6. **Assembly language and object code**
   a) In programming, what is an assembly language? What is its role in a toolchain? [W↗]
   b) What is the *x86* assembly language? What are its most used syntaxes? Treat an example. [W↗]
   c) How is an object code produced from an assembly code? Treat an example in *C*. [W↗]

7. **Linking and building**

   a) In a toolchain, what is a linker? Describe its purposes and results. [W↗] [↗]

   b) What is the GNU linker? Present it and find out if such a utility exists in *minGW*. [W↗] [↗]

   c) What is the `make` command? Present it and find out if such a command is available on *Windows*. [W↗] [↗]

8. **Static libraries**

   a) In programming, what is a library? Explain the interests and main concepts of this topic. [W↗]

   b) Specifically, what is a static library? Explain how it works within the tool chain. [W↗]

   c) How is a static library created and managed in *Arduino* IDE? [↗]

9. **Dynamic libraries**

   a) What are the principles of dynamically linked libraries – also known as shared libraries? What are their main advantages over static libraries? [↗]

   b) How is it possible to create a dynamically linked library with a *C* language toolchain? [↗]

   c) Are dynamic libraries used by operating systems? What is the allegedly *hell* about them? [W↗]

10. **Runtime environment**

   a) In general, what is runtime, and a so-called runtime environment? [W↗] [W↗]

   b) Before runtime, what is the role of the program loader? [W↗]

   c) What is a runtime library? [W↗] Which one is mainly used for *C* language programs? [W↗]