Exercises:

1. Create a program that converts decimal integral numbers into binary numbers and binary numbers into decimal integral numbers. – The program should have a menu etc. as well.
2. Learn how you can get a memory view of a program running in the debugger.
   a) Which presentation of negative values (e.g. the -1) does your machine use in memory? If needed use screenshots to show this.
3. What happens in your IDE if you divide by 0 in debug mode? What happens if you divide by 0 in a release-compiled executable on the console (w/o IDE)?
4. Write a function that rounds a double value to an int value. Prove its functionality with some unit tests.
5. Write a program that proves the "end of the precision" of the datatype float. How many digits can be represented "correctly"? If needed use screenshots to show this.

Remarks:

- Everything that was left unspecified can be solved as you prefer.
- In order to solve the exercises, only use known constructs, esp. the stuff you have learned in the lectures!
- **Please obey these rules for the time being:**
  - **The usage of goto, C++11 extensions, as well as #pragmas is not allowed.**
  - **The usage of global variables is not allowed.**
  - **You mustn't use the STL, esp. *std::string*, because we did not yet understood how it works!**
  - **But *std::cout*, *std::cin* and belonging to manipulators can be used.**
  - **You mustn't use new and delete!**
  - **You are not allowed to use C++ references instead of pointers.**
- Avoid magic numbers and use constants where possible.
- The results of the programming exercises need to be runnable applications! All programs have to be implemented as console programs.
- The programs mustn't have any memory leaks!
- The programs need to be robust, i.e. they should cope with erroneous input from the user.
- You should be able to describe your programs after implementation. Comments are mandatory.
- In documentations as well as in comments, strings or user interfaces make correct use of language (spelling and grammar)!
- Don't send binary files (e.g. the contents of debug/release folders) with your solutions! Do only send source and project files.
- Don't panic: In programming multiple solutions are possible.
- If you have problems use the Visual Studio help (F1) or the Xcode help, books and the internet primarily.
- Of course you can also ask colleagues; but it is of course always better, if you find a solution yourself.