Exercises and assignments, C-programming

**Work package nr 1 / Intro week**

**Exerc_1_ 0**

Install (or test an existing) an IDE to be used when working with the exercises below and for the project in the project course running parallel with this course.

For the project course there is a proposal for using Eclipse , gcc, gdb. During this course, I recommend to use Visual Studio or Visual Studio Code and gcc. You should install and test such an environment on some of the group's computers.

In this course we are going to develop rather small programs, so it doesn't matter what IDE you will use. Good examples are Eclipse or CodeLite. You should also learn to work directly with the compiler (gcc) together with a simple editor (Notepad++, Visual Studio Code) from the console window.

A good programming environment, however, offers a lot more features, e.g. integrated debugger. Therefore I will use Visual Studio in the course.

When you install the IDE, please write a short program to test it. A good example is a program which prints the string "Hello World!" on the console.

**Exerc_1_ 1 :** (Filename exerc_1_1.c)

Write a program that reads in an integer number between 1 and 5 from the keyboard and prints out one of existing five sentences on the console depending on what number was entered.
The program continues to ask for a new number and exits if number isn´t in the interval 1 to 5.

Exercises and assignments, C-programming

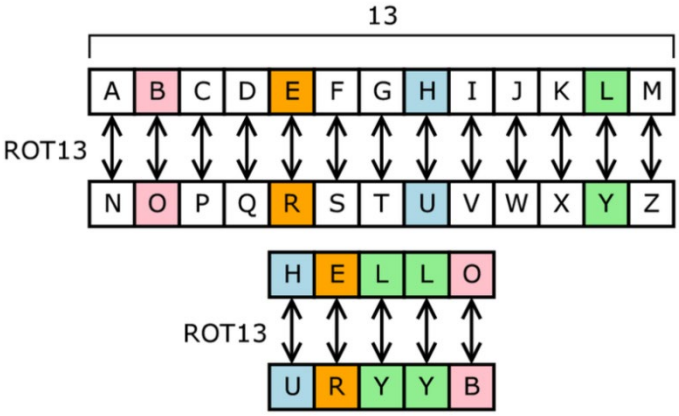
**Exerc_1_ 2 :** (Filename exerc_1_2.c)

Write a program that reads in a sentence of MAX characters and counts the number of words in it. The number of words should then be printed out on the console window.

Exercises and assignments, C-programming

**Exerc_1_ 3: Encryption ,** (Filename exerc_1_3.c)

Create a very simple encryption program. The program is based on the principal of “shifts of characters” in the ASCII-code table. In the example below, A has shifted to N, B to O, etc., that will mean 13 steps in the table. Only capital letters are viewed in the figure but the same ideas applies to lower case letters. The word HELLO becomes URYYB after encryption.




The user enters a text and the program prints out the encrypted text. Let the program read character by character, and encrypt it as above. The program is repeated until EOF indicated the program ends. (EOF, the user enters Ctrl +z for Windows and Ctrl + d for Linux system).

Example of a test run :

*HELLO (+enter)*
URYYB
*Banana (+enter)*
Onanan
*( +Ctrl-z)*
(Program ends)

Exercises and assignments, C-programming

**Exerc_1_ 4** : **Guess the number** , (Filname exerc_1_4.c)

You should develop a very simple game in which the computer creates a random integer number between 1..100. The user then tries to guess the number. The program should work as specified below:

- The computer creates a random number
- The user guess the number
- The computer respond by printing one of :
  You have guessed xx times and your guess is correct. Or
  Your guess is to low or to high.
- If wrong the user is asked for a new guess, this will continue until the guess is right or the number of guesses exceeds the value MAX_NUMBER.
- After end of one round the user is asked for a new round or to finish.

The program should only except guessed numbers in the range of 1 …100.

An option, but not a demand, is to secure that the program not fail (crashes) if a user by accident put in any character instead of a number.

---