# Exercise 3

## 3.1 Truth Tables (Paper Exercise)

Please consider the following logical expression: $!a\&\&b$, where $a$ and $b$ are 0 or 1. One can create a truth table of these variables:

| a | b | $!a\&\&b$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

The first and second column hold the variables a and b, and all the possible combinations (there are $2^n$, where $n$ is the number of variables). The third column holds the RESULT of the logical operation ($!a\&\&b$).

Remember what the logical operations do:

- $!a$ gives the "opposite of a", so the truth table of $!a$ is

| a | $!a$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

- $a\&\&b$ gives 1 only if both $a$ and $b$ are 1, else gives zero (Truth table is exercise 1.1)
- $a||b$ gives one if one or two of a and b are 1 (Truth table is exercise 1.2)

Please write the truth tables for the following logical expressions.

1. $a\&\&b$ ( a and b)
2. $a||b$ ( a or b)
3. $!a\&\&!b$
4. $!(a||b)$ (and compare to 3). This will show you De Morgan's law
   → De Morgan's law: $!(a||b) =!a\&\&!b$, $!(a\&\&b) =!a||!b$
5. $!a\&\&(b||c)$
6. $b||(!a\&\&c)$

## 3.2 Square-root

Consider, compile and run the program below:

- What does it do?
- What is the problem with it? (mathematically)

Improve the program in the following way:

- Let the user enter the number of which the square-root should be taken of. (use "scanf")
- Use a condition (if) to solve the problem of the program. The program should inform the user that the entered number was smaller than zero and that the squareroot of such a number cannot be calculated in R.

```c
#include <stdio.h>
#include <math.h>

int main()
{
        float a;
        float b;

        a=103;
        b=-78;

        float wurzela = sqrt(a);
        float wurzelb = sqrt(b);

        printf("%f\n",wurzela);
        printf("%f\n",wurzelb);


        return 0;
}
```

## 3.3 Quadratic equations

$$ax^2 + bx + c = 0$$

Write a program that solves this quadratic equation. The user must be able to enter a, b and c into the program (”scanf”).

- Do you remember the formula to solve quadratic equations?
- What has to be considered mathematically?
- Try to formulate the program first on paper in pseudo code.
- Write the program in C
- Make the program user-friendly by always telling the user what happened (e.g. “With the parameters you entered, the quadratic equation only has one solution: 0”).

## 3.4 Dates

Write a program that requires two dates as input in the format DD.MM.YY and says which is the more recent, or if they are equal. Hint: scanf can also read multiple variables a time:

```c
scanf("%d.%d.%d",&d,&m,&y);
```

This line means, scanf is expecting 3 integers separated with dots, and puts them into variables d, m and y. The same applies for printf:

```c
printf("%02d.%02d.%04d \n",d,m,y);
```

will print the the date, formatting it to XX.XX.XXXX.

- if you want, you can extend the program to check the validity of the dates (it is up to you how smart, e.g. considering leap-years, the solution is)

## 3.5 Repeat numbers

Write a program in which the user can input 2 numbers. The first number is the number that is repeated, and the second number how many times it is repeated. Example: f(9,8) → 99999999. (It doesn't have to work for the number 0 as first parameter, though.)

- This problem can be solved without loops. Try to find a mathematical way to do this.
- After you have found the mathematical solution, think about forbidden input values (for example those that cause a division by zero), and intercept them using "if". Don't forget to tell the user what he has done wrong.

## 3.6 Exercise, day of the week

Write a program that requires an input date DD MM (year is 2020) and says which day of the week it is.