

Exercise 6: Functions

FAQ link:

<https://docs.google.com/document/d/1XKNrv16vpfOv9wegMcqkhkZnxrrQFpc29A5gFgNsa5g/edit?usp=sharing>

6.1 Exercise 1

- Write a function `sign(float x)`, which returns

```
-1 if x is negative
0  if x is zero
1  if x is positive
```

- Think about what type this function should return. (int or float? what makes sense?)

6.2 Exercise 2

- Write a function `factorial(n)` which returns $n!$ (Fakultaet)
- Think about what type this function should return. (int or float? what makes sense?)
- Also think about what data type the parameters expect. (For example does it make sense to pass a float value to the fakultaet function?)

6.3 Exercise 3

- Think about how you would implement a squareroot function. Think about how precise you want to be.
- Implement the function in C now. I recommend defining the function like this: (Because we do not want to interfere with the built-in `sqrt` function)

```
float mysqrt(float ofnumber, float precision)
```

You can see that in your version of `sqrt`, the user can also say how precise the value should be. (Hint: After thinking by yourself, try to search for “Babylonian method”)

– From here on, Exercises are general. This means that you will need everything you have learned so far in order to solve them. –

6.4 Exercise 4

Consider the following problem. We have 4 weather stations, 1: Bern, 2: Zurich, 3: Geneva, 4: Basel. These weather stations send their data (Temperature and relative humidity) to a server somewhere in Switzerland. Due to the conditions, the data not always reach the server. The data from the server look the following way:

```
720 3
$1: minute ID: 1 is 00:01, 2 is 00:02, ..., 720 is 12:00
$2: number of measurements that reached the server
```

Listing 1: header line

```

1 25.2 60.2
3 23.1 59.7
4 22.1 63.1
$1: station number, here: Bern, Geneva, Basel
$2: temperature
$3: relative humidity

```

Listing 2: data lines (timestamp)

so there is always an entry with the timestamp of the measurement and the number of stations that reported a measurement (ranging from 0 to 4) please find this file online: [weather.txt](#)

The task is now to do the following:

- The program should read the file
- Print the average temperature and rel. humidity of each station. Also print the standard deviation.
- Print the efficiency for the data transfer for each station (efficiency = $(np/nt)*100$, where np is the number of times that the station succeeded in transferring data, nt is the total number of trials)

6.5 Exercise 5, A Game

- Write a program that picks an integer value between 1 and 1000 randomly (see below how to create a random value).
- The game shall then ask the user to guess the number. Depending on the number the user entered, the program should tell the user if the number is too high, too low, or correct (if you want to make the game easier, you can also say "much too high" etc. this is up to you).
- The program terminates as soon as the correct number is found.

How to create random numbers: include those header files at the beginning of your c code,

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

```

at the beginning of your main function, do:

```

srand(time(NULL));

```

(this code sets the so called random seed according to the current time. We must do this in order to prevent the program from always using the same random numbers.) if you want to get a random number in the range (0,1000) anywhere in the program, you can use `rand()`, like this:

```

int getrand;
getrand = rand()%1000;

```

`rand()` returns a number between zero and `RAND_MAX`. `RAND_MAX` depends on the system, $2^{32} - 1$ on Linux, $2^{12} - 1$ on Windows.