Exercise 4 - Arrays and Structs

This weeks exercise sheet will focus on implementing some functions on `arrays` and `structs`.

If you have any questions or problems, please write in the Moodle!

Exercise 1 — Arrays

For this Exercise write your functions into `Exercise1_Arrays/Arrays.cpp`*.*
Try out your code as usual in `main.cpp`*. You can have a look into* `Exercise1_Arrays/Arrays.h`*, if you are having problems figuring out your function headers.*

a) *Given the Array* `grades` *and its length* `amount_of_grades`*. Write a function* `average` *which, given an array* `double[]` *and the arrays length as* `int`*, returns the average of the array as* `double` *value.*

   The average of a given array can be calculated with the following formula:

$$\frac{1}{n}\sum_{k=1}^{n} a_k \quad \text{Example: } \frac{3+2+1}{3} = 2$$

b) *Now we want to implement a* `contains` *function checking if a given number is an element of a given array or not. Since we learned, that we don't want to compare two* `double` *values with* `==` *, we are going to write a* `compare` *function first.*

   1) *Our* `compare` *function shall take two* `double` *values as parameter and return* `true`*, if the distance between the two numbers is lower than our threshold* `0.00001` *and otherwise return* `false`*.*

      For example the numbers 5 and 4.9 are given:

$$|5.0 - 4.9| < 0.00001$$
$$0.1 < 0.00001$$
$$\Rightarrow \text{False}$$

   2) *Implement the function* `contains` *which, given an array* `double[]`*, the arrays length as* `int` *as well as a* `double` *target, returns the index containing the target or, if the target can not be found, returns* `-1`*. Use your* `compare` *function when comparing two* `double` *values.*

   3) ***Bonus-Task:***

      Compare your solution to the function `contains_binary_search`*.*

Do they do the same thing? How many memory-cells are checked, before the right one is found? What assumption must be fulfilled, for `contains_binary_search` *to work properly?*

You will also find answers to these questions, when looking up `binary-search` inside of your web-browser.

Exercise 2 — Structs

For this Exercise write your methods into `Exercise2_Structs/Structs.h` *Try out your code as usual in* `main.cpp`*.*

a) *Write a* `Struct` *declaration* `Pizza` *with the* `float`*-fields* `diameter` *and* `price` *simular to the one from the lecture. Inside the* `main` *function, initialize two Pizzas taking values from your trusted Pizza-Delivery(Or make up values yourself).*

b) *Write the method* `price_per_cm2`*, which returns exactly that as* `float`*-value.*

c) *Write the method* `five_euro_get_you_x_cm2`*, which is using the* `price_per_cm2` *method to calculate the amount of cm2 you get for 5€ and returns that as* `float`*-value.*

Test your methods via the `main` *function, so see which Pizza you should be ordering more often.*