

## Exercise Sheet 6

**Warmup (optional):** Find and correct the problems in the following code fragments:

- a) 

```
void func(int a)
{ return a*a; }
```
- b) 

```
double func(double x)
{ printf("test value: %f\n", x); }
```
- c) 

```
double func(double x, y)
{ return x*y; }
```
- d) 

```
int func(long n)
{ n = 10*n; }
```

**Exercise 6.1.** Write the prototypes of the following functions?



- a) The function `median3()` returns the mean value of three double values passed as arguments.
- b) The `logStatus()` function writes the current time and status of the program to a log file. The function has no parameter and no return value.
- c) The function `slope()` returns the slope of a straight line through two points in the plane. The coordinates of the two points  $x_1, y_1, x_2, y_2$  are passed to the function as double values.
- d) The function `gcd()` determines the greatest common divisor of two integers passed as arguments.
- e) The function `geomSeries()` returns the  $n$ -th element  $s_n$  of a normalized geometric series, i.e. the value  $s_n = 1 + q + q^2 + \dots + q^n$ . The function receives the floating point number  $q$  and the integer  $n$  as arguments.
- f) The `initApplication()` function implements the startup sequence for some program. It receives no argument and returns `true` if the initialization was successful, otherwise `false`. The constants `true` and `false` are of type `bool`, which is defined together with the constants in the standard header file `stdbool.h`.

**Exercise 6.2.** Each point  $P$  in three-dimensional space is uniquely determined by its Cartesian coordinates  $(x, y, z)$ , where  $x, y$  and  $z$  are real numbers. A point  $P$  also represents the array from the origin  $(0, 0, 0)$  to the point  $P$ . Define a suitable structure `Point3D` to represent a point in three-dimensional space. Then create the following functions for objects of this type:

- a) The function `sumP3D()` adds two points and returns the sum as return value. The sum  $P_1 + P_2$  of two points  $P_1 = (x_1, y_1, z_1)$  and  $P_2 = (x_2, y_2, z_2)$  is defined as  $(x_1 + x_2, y_1 + y_2, z_1 + z_2)$ .

- b) The function `toStringP3D()` function shall return a pointer to a `static` string containing the coordinates of a point in the form  $(x, y, z)$ . When called again, the string of the previous call is overwritten! Take care to allocate enough space in the string to take up the coordinates.
- c) Try assigning your point structures directly to each other (to copy the coordinates of one point to the other). How does the situation change if you do not assign structures, but *pointers* to your structures?
- d) Try changing `struct` into `union`. Would your code still work? Why (not)?

*Hints:*

- Use the `sprintf()` function to write data formatted into a string. The first argument is the destination string. The other arguments are the same as `printf()`.
- Put the structure definition and the functions as functions into a header file. A template header file `point3d-template.h` that you should use is attached.   
Be careful to pass pointers to your points as arguments to the functions (see the prototypes in the provided template).
- A template `point3d-template.c` is attached.   
You must leave the file names unchanged (for the automatic testing).