Exercise 8: Structure

8.1

Create a structure Vector3D. Define functions to calculate dot-product (skalarprodukt) and cross-product (vektoprodukt). You may reuse the code you have already written a few weeks ago (Exercise 5.2), but put it inside functions which take Vector3D types as input parameters and return the appropriate type.

8.2

Consider the following structure:

```
struct record {
  unsigned int date;
  char firstname[30];
  char lastname[30];
  unsigned int year;
  char nation[30];
  float time;
  float wind;
};
```

As you can see, the first name, the last name and the nation are strings and they can record up to 29 characters (the last one is always needed to store the '\0' character).

• Write a program which reads data from a list of 100 m world record holders (create a file "records.txt" and copy-and-paste the following data), then stores the information inside an array of records and finally prints the information of everybody to the screen.

```
1968    Wyomia Tyus                 1945    United_States   11.07   1.2
1972    Renate Stecher              1950    East_Germany    11.07   0.2
1976    Inge Helten                 1950    West_Germany    11.04   0.6
1976    Annegret Richter            1950    West_Germany    11.01   0.6
1982    Marlies Oelsner-Goehr       1958    East_Germany    10.81   1.7
1984    Evelyn Ashford              1957    United_States   10.76   1.7
1988    Florence Griffith-Joyner    1959    United_States   10.49   4.7
```

Listing 1: records.txt

• Print the information of the youngest person to set a world record. Note that "date" is the year of the competition, while "year" is the year the athlete was born.

• Assume that the tail wind in each record made the athlete faster by exactly 10% of the measured velocity (the wind is measured in m/s). Calculate the "corrected" times the athletes would have had if there had been no wind during the race. Would the overall ranking change?

8.3

Use the source code below which contains three different structs: Sphere, Cuboid (Quader), Cylinder. We have also written the function prototypes to calculate the shape's volumes. Implement these functions, and test their functionality.

```c
#include <stdio.h>

//struct definitions
typedef struct {
  float radius;
} sphere;

typedef struct {
  float side_x;
  float side_y;
  float side_z;
} cuboid;

typedef struct {
  float height;
  float radius;
} cylinder;

//function prototypes
float volume_sphere(sphere);
float volume_cuboid(cuboid);
float volume_cylinder(cylinder);

//main program
int main() {
  sphere a;
  cuboid b;
  cylinder c;

  a.radius = 10.2;

  b.side_x = 2.3;
  b.side_y = 7.3;
  b.side_z = 7.7;

  c.radius = 3.4;
  c.height = 5.6;

  float sp_vol = volume_sphere(a);
  printf("sphere volume: %f\n" , sp_vol);

  float cb_vol = volume_cuboid(b);
  printf("cuboid volume: %f\n" , cb_vol);

  float cy_vol = volume_cylinder(c);
  printf("cylinder volume: %f\n" , cy_vol);

  return 0;
}

//function implementations

float volume_sphere(sphere the_sphere){
  //implement me
}

float volume_cuboid(cuboid the_cuboid){
  //implement me
}

float volume_cylinder(cylinder the_cylinder){
  //implement me
}
```