02635 Fall 2016 — Module 5

Homework

- Read chapter 11 pp. 429-461 in "Beginning C"
- Read chapter 6 in "Writing Scientific Software"

Exercises

1. Do exercise 11-1 in "Beginning C"

2. Take this quiz to test your understanding of *structures*

3. Write a short program that (i) prompts the user to enter three points in $\mathbb{R}^2$ that define a triangle, and (ii) computes and prints the area of the triangle. Your program should use structures and functions.

   Hint: Define a structure that represents a point in $\mathbb{R}^2$ and another structure that represents a triangle. Write a function that takes a triangle structure as input and returns its area.

4. Extend your code from exercise 3 with a function that can check if a point is inside a triangle. Write a program to test it.

   Hint: To check if a point (x, y) is inside a triangle with vertices (x_1, y_1), (x_2, y_2), and (x_3, y_3), if suffices to check that the so-called barycentric coordinates of (x, y) are nonnegative. Given the vertices (x_1, y_1), (x_2, y_2), and (x_3, y_3), the barycentric coordinates of (x, y) are given by

$$\begin{aligned}
\lambda_1 &= \frac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)} \\
\lambda_2 &= \frac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)} \\
\lambda_3 &= 1 - \lambda_1 - \lambda_2.
\end{aligned}$$

5. Extend your code from exercise 3 with a function that can check if two triangles are conguent. Write a program to test it.

   Hint: See Congruence of triangles.

6. Consider the following structure declaration:

```
struct my_struct {
   int i;
   short j;
   char c;
};
```

Write a program with a variable of type `struct my_struct` and print out the size of each of the fields (`i`, `j`, and `c`) as well as the size of the struct itself. Do the sizes of the three fields add up to the size of the struct?

Hint: Use the `sizeof` operator.

Optional exercises

1. A polynomial of degree n

$$f(x) = \sum_{i=0}^{n} a_i x^i, \quad a_n \neq 0$$

can be represented by means of a structure

```
struct polynomial {
    int degree;
    double *coef;
};
```

where the field `degree` corresponds to the degree n, and `coef` is a pointer to an array of length $n+1$ with the $n+1$ coefficients $a_0, \ldots, a_n$. Write a function that can multiply two polynomials of arbitrary degree, and write a short program to test it.

2. Skim through Wikipedia: List of data structures to learn about common data structures.