Dirk Praetorius,
Thomas Führer,
Michele Ruggeri

Wintersemester 2014/15
14.10.2014

Übungen zur Vorlesung
Einführung in das Programmieren für TM

Serie 3

Aufgabe 3.1. Write a `void`-function `divisor` which checks if a given number $x \in \mathbb{N} := \{1, 2, 3, \dots\}$ is divisible by 2, 3, or 6. Additionally, write a main program that reads in the number x, then calls the function `divisor`, and prints out the result.

Aufgabe 3.2. Let the three points (x, y), (u, v), and (a, b) in $\mathbb{R}^2$ be given. Write a function `points` which checks if the three points lie on the same line. Additionally, write a main program which reads in the six values and prints out the result on the monitor.

Aufgabe 3.3. Write a `void`-function `triangle`, which, given three edge-lengths $a, b, c \in \mathbb{R}$ with $a, b, c \geq 0$, determines if the resulting triangle is equilateral, scalene, isosceles, one-dimensional degenerate (the sum of two edges equals the third one) or impossible (the sum of two edges is smaller than the third one). Then, write a main program which reads a, b and c from the keyboard and call the function `triangle`. Save your source code as `triangle.c` into the directory `serie03`.

Aufgabe 3.4. Write a `void`-function `sort3` which gets three real numbers $x, y, z \in \mathbb{R}$ as input. Furthermore, the numbers should be printed out in descending order. Additionally, write a main program that reads in the numbers x, y, z and calls the function.

Aufgabe 3.5. Write a `void`-function `lines`, which characterizes the mutual position of two lines: Given six parameters $a, b, c, d, e, f \in \mathbb{R}$, the equations

$$\begin{aligned} ax + by &= c, \\ dx + ey &= f \end{aligned}$$

define two lines in the plane. The function `lines` determines if the lines defined by the six input parameters $a, b, c, d, e, f \in \mathbb{R}$ are *parallel*, *coincident* or *intersecting*. In the third case, the function computes and prints out the coordinates of the intersection point. Then, write a main program which reads the six parameters a, b, c and d, e, f from the keyboard and calls the function `lines`. Save your source code as `lines.c` into the directory `serie03`.

Aufgabe 3.6. Implement the following game. Compute a random number between 0 and 15. You have three tries for guessing the right number. If your first or second guess is wrong, the program should tell you if the number you have entered is larger or smaller than the correct one. If also your third (and last) guess is wrong, then the right number should be printed on the monitor. Random numbers can be created as follows: First include the header files `stdlib.h` and `time.h` into your program. The following code lines

```
srand( (unsigned) time(NULL) );
int randnumber = rand() % 16;
```

generate a random number between 0 and 15. The variable `randnumber` has the type `int`.

Aufgabe 3.7. The Fibonacci series is recursively defined by $x_0 := 0$, $x_1 := 1$, and $x_{n+1} := x_n + x_{n-1}$. Write the function `fibonacciRec` which returns x_n for given n.

Aufgabe 3.8. Recall the meanings of the terms *Lifetime & Scope*. What is the output of the following code lines?

```c
#include <stdio.h>

int max(int,int);

main() {
  int x = 1;
  int y = 2;
  int z = 3;

  printf("(x,y,z) = (%d,%d,%d)\n",x,y,z);

  {
    int x = 100;
    y = 2;
    z = max(x,y);
    printf("(x,y,z) = (%d,%d,%d)\n",x,y,z);

    {
      int z = y;
      y = 200;

      printf("(x,y,z) = (%d,%d,%d)\n",x,y,z);
    }
    printf("(x,y,z) = (%d,%d,%d)\n",x,y,z);
  }
  printf("(x,y,z) = (%d,%d,%d)\n",x,y,z);
}

int max(int x, int y) {
  if(x>=y) {
    return x;
  }
  else {
    return y;
  }
}
```

Draw a timeline and visualize the lifetime and the scope of the variables `x,y,z`. Moreover, mark all blocks and functions.