Valentin Helml
Dirk Praetorius
Julian Streitberger

Summer term 2022
25 March 2022

Exercise
Scientific programming in mathematics

Series 4

Exercise 4.1. Write a C program that reads two integers `a` and `b` $\in \mathbb{N}$ from the keyboard, computes the mean $(a+b)/2 \in \mathbb{R}$, and prints the result to the screen. Save your source code as `mean.c`.

Exercise 4.2. Consider a square with length $L > 0$ and vertices $(0,0), (L,0), (L,L)$ and $(0,L)$. Moreover, consider a point $(x,y) \in \mathbb{R}^2$. Write a program `locate.c`, which reads $\mathtt{L} > 0$ and $\mathtt{x}, \mathtt{y} \in \mathbb{R}$ from the keyboard and prints to the screen the position of the point $(\mathtt{x}, \mathtt{y})$ with respect to the square. Note that a point can be (strictly) inside the square, on its boundary, or outside of it.

Exercise 4.3. Write a program `sort3.c` which reads three real numbers $\mathtt{x}, \mathtt{y}, \mathtt{z} \in \mathbb{R}$ from the keyboard and prints to the screen the numbers in descending order. This means that the maximum $\max\{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$ is printed at first and the minimum $\min\{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$ at last.

Exercise 4.4. Write a `void` function `points` which, given three points $(x,y), (u,v)$ and $(a,b) \in \mathbb{R}^2$ checks wether they lie on the same line and prints the result to the screen. Moreover, write a main program which reads the six coordinates of the points from the keyboard and calls the function. Save your source code as `points.c`.

Exercise 4.5. Write a recursive function `binomial` which computes and returns the binomial coefficient $\binom{n}{k}$ of two given integers $0 \le k \le n$. Use the addition formula

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1} \quad \text{for } 1 \le k < n$$

with $\binom{n}{0} = 1 = \binom{n}{n}$. Write a main program `binomial.c`, which reads `k` and `n` from the keyboard and prints to the screen the result $\binom{\mathtt{n}}{\mathtt{k}}$.

Exercise 4.6. Write a recursive function `division`, which computes and returns the result of the integer division m/n (division without remainder) for two given integers $m \ge 0$ and $n > 0$. The function may only use the arithmetic operation `+` and `-`. Moreover, write a main program `division.c` which reads `m` and `n` from the keyboard and prints the result `m/n` to the screen.
Hint: For any $y \neq 0$, it holds that $x/y = 1 + (x-y)/y$.

Exercise 4.7. The Fibonacci sequence is recursively defined via $x_0 := 0$, $x_1 := 1$ and $x_{n+1} := x_n + x_{n-1}$. Write a recursive function `fibonacci`, which computes and returns x_n for given $n \in \mathbb{N}_0$. Then, write a main program `fibonacci.c`, which reads $\mathtt{n} \in \mathbb{N}$ from the keyboard and prints the value of x_n to the screen.

Exercise 4.8. According to an old legend, there was a temple in Hanoi, which contained a large room with three time-worn posts in it surrounded by 64 golden disks of different diameters. When the temple was built, the disks were arranged in a neat stack in ascending order of size on the first rod, the smallest at the top, thus making a conical shape. Since that time, the temple priests have been moving the disks with the objective of moving the entire stack to the third rod (preserving the ascending order). The second rod is auxiliary. All disk movements must be in accordance with the following immutable rules:

- Only one disk can be moved at a time.
- Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack, i.e., a disk can only be moved if it is the uppermost disk on a stack.
- No disk may be placed on top of a smaller disk.

According to the legend, when the last move will be completed, the world will end.
The task can be accomplished with the use of a recursive algorithm. Let n denote the total number of disks ($n = 64$ in the legend). In order to move the upper $m \leq n$ disks located on the i-th rod to the j-th rod ($i, j = 1, 2, 3$), proceed as follows:

- Move the upper $m - 1$ disks from the i-th rod to the k-th rod with $k \neq i, j$.
- The largest of the m disks is now on top of the i-th rod and can be moved to the j-th rod.
- Finally, the $m - 1$ disks from Step 1 can be moved from the k-th rod to the j-th rod.

The choice of $m = n$, $j = i$ and $j = 3$ in the algorithm solves the priest task. Write a recursive function `void hanoi(int m, int i, int j)`, which implements the algorithm. Any single movement of any disk must be printed out to the screen, e.g., `Move a disk from rod 2 to rod 3.` Furthermore, write a main program `hanoi.c` that reads `n` from the keyboard and prints out the list of all movements. To test the algorithm, use $n \ll 64$, e.g., $n = 3, 4, 5$.