

## Scientific Computing 1 2nd Homework

**Handout:** 18<sup>th</sup> Oct. 2016

**Return:** 26<sup>th</sup> Oct. 2016

---

*“When reading the code in about six months and asking yourself: who wrote this crap?  
The answer should not be: YOU!”*

Basically that means:

- Try to always use meaningful names for functions, variables, ...
  - Write documentation wherever necessary.
  - Use indentation to increase readability of the code.
  - Add a short statement describing its purpose and basic behavior to each function.
  - ...
- 

### Exercise 1:

(4 Points)

Write a C program which reads two integers  $a$  and  $b$  and computes  $\frac{a}{b}$ .

- Find out what happens if  $b$  is zero.
- Does the compiler recognize if there is a hard-coded division by zero?
- Rewrite the program to floating point numbers. What happens now if  $b$  is equal to zero?
- Modify the program such that  $b = 0$  is detected and avoided before an error occurs.
- Is the modulo-operator also affected?

### Exercise 2:

(4 Points)

Write a C function which converts a temperature given in degrees Fahrenheit to degrees Celsius. The conversion is done with

$$T_C = (T_F - 32) \cdot \frac{5}{9}.$$

Demonstrate the function with two examples:

- Read a temperature in degrees Fahrenheit from the standard input and print out the corresponding degrees Celsius.

- b.) Read a lower and a upper bound from the standard input defining an interval in degrees Fahrenheit. Print a table containing the temperatures in degrees Fahrenheit and degrees Celsius to the screen. In the table use steps of 1 Fahrenheit.

**Exercise 3:** (2 Points)

Write a C program which reads a positive floating point number from the standard input and rounds it correctly to the nearest integer. Do not use any sophisticated routine from C's math library for this.

**Exercise 4:** (4 Points)

Write a C program which computes the prime factorization of a positive integer. The integer is read from the standard input and the result is printed on the screen. If a factor occurs more than one time write it as a power. For example, the program should work like:

```
Insert a positive number: 92
The prime factorization of 92 is:
2^2 * 23^1
```

**Exercise 5:** (5 Points)

Write a C function which computes the value of the exponential function  $y = \exp x$  by using the infinite sum

$$\exp(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

The summation should stop if either a previously given maximum number of summands have been added or the difference between two successive summands is below a given tolerance  $\tau$ . Think of an efficient evaluation of the summands. Point out problems that can occur in your implementation.

**Exercise 6:** (3 Points)

Write a C program which reads one integer value from the standard input and reverses the order of the digits. For example, if the user inputs 4711 the output must be 1174. The procedure should be realized using integer arithmetic only.

**Overall Points: 22**