Otto-von-Guericke-University Magdeburg
Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods in Systems and Control Theory

Dr. Jens Saak, Dipl.-Math. Martin Köhler
Website: http://www.mpi-magdeburg.mpg.de/csc/teaching/16ws/sc1/

Winterterm 2016/2017

Scientific Computing 1
2nd Homework

Handout: 10/18/2016 **Return:** 11/04/2016

*"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.
- ...

Changes: Remove the old Exersice 8

Exercise 1: **(4 Points)**

Implement Euclid's algorithm to compute the greatest common divisor of two integers as a C function. Derive a second function which computes the least common multiple of two given integers. Demonstrate the usage of both functions in a small program.

Exercise 2: **(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 postive number: 92
The prime factorization of 92 is:
2^2 * 23^1
```

Exercise 3: **(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 τ. Think of an efficient evaluation of the summands. Point out problems that can occur in your implementation.

Exercise 4: (5 Points)

Design a data structure which represents a point in $\mathbb{R}^3$. On top of this structure develop the following functions:

a.) A function which reads three floating point numbers from the standard input (as (x, y, z) coordinates of the point) and return them as an instance of the previously defined structure via `return`.

b.) A function which reads three floating point numbers from the standard input (as (x, y, z) coordinates of the point) and passes the values back via a pointer in the parameter list. The return value of the function should be `void`.

c.) A function which has two points as input parameters and returns their euclidean distance.

Demonstrate all function in a small C program.
Explain the difference between how the structure is handled in Function a.) and Function b.).

Exercise 5: (3 Points)

Write a C program which reads a positive floating point number from the standard input and rounds it correctly to the nearest integer. Hint: A typecast to `int` simply truncates the decimal places.

Exercise 6: (6 Points)

Write a C program which analyzes some measured data. The program should behave as follows:

- The user should enter the total number of measurements.
- The program asks the user for every single measured value.
- The minimum, the maximum and the average of all values are determined and printed to the screen.

We assume that the measured values are floating point numbers.
Hint: The number of values is not known during the development or compile time.

Exercise 7: (5 Points)

We consider an array `int *f` of `n` integers. Write a program which reads the array from a file containing one integer per line. The first entry is the total number of integers to read.
Analyze the array and determine the two indices $i, j \in \{0, n-1\}, i \leq j$ such that

$$S_{ij} := \sum_{k=i}^{j} \texttt{f[k]}$$

is maximized. Think about an efficient solution.
Example data sets are available on the lectures website.

Example: Consider the following array of length 10:

Index	0	1	2	3	4	5	6	7	8	9
Value	-1	3	4	-2	5	1	-9	4	2	-2

Then the maximum of S_{ij} is $S_{15} = 11$ beginning at $i = 1$ and ending at $j = 5$.

Exercise 8: **(4 Points)**

`Makefiles` support the developer to build large projects easily. Write a `Makefile` which has a target for each source code of the homework. By calling `make all` all programs should be compiled and also calling `make clean` should clean up all binary files created by the make process.

Overall Points: 36