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/18ws/sc1/

Winterterm 2018/2019

Scientific Computing 1
3rd Homework

Handout: 25th Oct. 2018 **Return:** 2nd Nov. 2018

*"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)

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: (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 3: (4 Points)

The BLAS library provides the function `ddot` to compute the scalar product of two n-dimensional vectors.

In most cases, like in the virtual machine, the BLAS library can be linked to the program by adding `-lblas`. Demonstrate the usage with two small vectors and verify the result.

Exercise 4: (6 Points)

We consider an array `int *f` of `n` integers. Write a function which analyzes the array and determines 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. The `main` function reads the array from a given file. The file is organized in the following way: The first line contains the number `n` of values stored inside the file. The following `n` lines contain one element of `f[]` per line.

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 5: (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.

Exercise 6: (2 Points)

You will get a C program from the previous exercise via e-mail. Take a look at it and comment it. Think about:

- Is the code readable or well formed?
- Is the purpose obvious?
- Are unclear statements documented?
- Are function and variable names meaningful?
- Are there parts which can be implemented better or more efficiently?
- ...

Overall Points: 25