Winterterm 2012/2013

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

Dr. Jens Saak, Dipl.-Math. Martin Köhler
Website: http://www.mpi-magdeburg.mpg.de/mpcsc/lehre/2012_WS_SC/

Scientific Computing 1
4th Homework

Handout: 10/24/2012 **Return:** 11/01/2012

Make sure you follow the basic rule:

> *"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: (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: http://www.mpi-magdeburg.mpg.de/mpcsc/lehre/2012_WS_SC/data/sum_data.tar.gz

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 2: (4 Points)

The BLAS library provides the function `ddot` to compute the scalar product of two n-dimensional vectors. Download this routine from http://www.netlib.org/blas/ and write a C interface for it. Demonstrate the usage with two small vectors and verify the result.

Exercise 3: **(5 Points)**

Write a function `min_element` which takes an integer array `f[]` and two indices i and j as input. The function has to return

$$\mathrm{argmin}_{i \leq k < j}\, \texttt{f[k]}.$$

Implement an easy sorting algorithm on top of `min_element` and the `swap` function from the lecture.

Demonstrate the algorithm the following way:

- Read an example file from Exercise 1.
- Sort the array.
- Write the sorted array to a file.

Exercise 4: **(5 Points)**

Write the following functions without using library functions from `string.h`:

a.) A function called `str_length` which takes a string as input and returns its length without the trailing 0-byte. Write a preprocessor macro `IS_NUL` which returns true if a given character is the 0-byte.

b.) A function called `str_find` which takes two strings as inputs and returns true if the second string is found in the first one or false otherwise.

Demonstrate both functions in a small main program.

Exercise 5: **(4 Points)**

`Makefiles` support the developer to build large projects easily. Write a separate `Makefile` for each of the previous Exercises on this sheet.

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 efficient?
- ...

Overall Points: 25