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
3rd Homework

Handout: 10/18/2012 **Return:** 10/25/2012

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)**

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 2: **(5 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:

a.) 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: **(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 4: **(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 5: **(5 Points)**

Design a data structure which represents a point in $\mathbb{R}^3$. Write a function which reads a point from the standard input and a function which computes the distance between two given points. Demostrate both functions in a small program.

Overall Points: 25