Prof. Dr.-Ing. Tim Güneysu
M. Sc. Florian Stolz
M. Sc. Moritz Peters

Ruhr-Universität Bochum
Faculty of Computer Science
Chair for Security Engineering

Winter Term 23/24

Computer Engineering 3 – Hardware-Oriented Programming

First Project

Submit answers until 2. November 23:59:59 via *Moodle* — Σ 100 Points

---

Your dear friend Maike B. gave you a hint to apply to LAME Inc., a relatively new and unknown enterprise, which hopes to shake up the vacuum robot market. As a former student of the Ruhr-University Bochum and because of your excellent mark in *Computer Engineering 3*, HR invited you and your friends immediately to an interview, where you have to prove that you can code. Below you find the functions HR asks you to implement.

Attention: No external library functions are allowed! All functions are located in `tasks/` and tests are provided in `main.c` . Comments may provide you with further instructions. Ask us if something is unclear.

Task 1: Prime Number Check (10 Points)

Write a prime number check, which utilizes *trial division*. This method is the most inefficient but also simplest one. It works like this: Simply check if the target number `x` is only divisable by itself and 1.

a) Implement `prime(unsigned int x)`, which whether the number is prime or not. Remember how `true` and `false` are represented in C! (10 Points)

Task 2: Reverse (30 Points)

Did you know that the word *racecar* reads the same backwords as well as forwards? Reversing strings manually however is a tedious task. Thankfully we can just let the microcontroller do this for us!

a) Implement the function `reverse(char* in, char* out, int len)`, which reverses the string `in` and stores at location `out`. (30 Points)

Task 3: Find the Sum (30 Points)

Here is a puzzle for you. Given a list of integers, we want to find two integers, which sum up to the number 2023. Once you found them, multiply them and return the result of the multiplication. For example: If the list would include 2020 and 3, then those sum up to 2023 and the result would be: 6060. For simplicity: The list only contains two such integers!

a) Implement the function `findSum(int* x, int len)`, which solves the puzzle! (30 Points)

Task 4: Square Roots (30 Points)

Unfortunately, processors cannot just calculate $\sqrt{x}$. Fortunately, we can approximate $\sqrt{x}$ using Heron's method. If you want to approximate the square root of x you perform the following steps:

1. Initially compute $y_0 = \frac{x+1}{2}$
2. Compute $y_1 = \frac{1}{2} * (y_0 + \frac{x}{y_0})$
3. Repeat step 2 with the updated value: $y_n = \frac{1}{2} * (y_{n-1} + \frac{x}{y_{n-1}})$ until you reach the precision you require!

a) Implement the function `sqrt(float x, int max_iterations)`, which calculates $\sqrt{x}$.
You *must* use a `while`-loop! (30 Points)