Data Processing on Modern Hardware
Assignment 4 – SIMD Vectorization

Handout: 24^{th} May 2023
Due: 31^{th} May 2023 by 14pm

Part 1: Characters to lowercase

In this task you have to implement a scalar and a vectorized version of the function toLower. This function takes an array of Ascii 'chars' and transforms all uppercase letters to lowercase letters. Test your implementation with different array sizes using the provided character generator.

Use the code skeleton provided in the gitlab repository[1]. Answer the following questions in your report:

1. Investigate whether GCC and clang can auto-vectorize the scalar implementation under different optimization settings (O3, O2, O1). You can use https://godbolt.org for your analysis.
2. Can you implement a branch-free scalar version? If yes, how good does it perform?
3. How large is the speed-up you can achieve with the vectorized version?

Part 2: Mandelbrot

You are given a scalar implementation of the Mandelbrot set[2]. Look at that scalar implementation and figure out how to vectorize the computation. The provided code skeleton creates a 'PPM' file that you can open to check the result of your vectorized implementation. Your tasks are:

1. Speed up the given code by creating a vectorized version of Mandelbrot using floats.
2. Try to maximize the performance by using the largest possible size of vectorization on your system.
3. Discuss the properties of the different versions of the code in your report.
4. Can you achieve an even wider vectorization by using fixed-point arithmetic instead of floating points?

Submission guidelines

This homework has a duration of one week. Fork the repository and commit your changes in the git.

The programming language of this homework is C. We provide you a simple code skeleton, feel free to add functions.

[1] https://gitlab.db.in.tum.de/data-processing-on-modern-hardware-2023

[2] https://en.wikipedia.org/wiki/Mandelbrot_set