Requirements:

This assignment requires you to have IBM POWER10 functional simulator installed. Refer to https://github.com/w-feng/CompArch-MIPS-POWER/blob/main/Tutorials/Tutorial-Power%2010%20Functional%20Simulator.pdf for the instructions on installing the simulator

---

Reference implementations

You may refer to the MMA best practices guide for more information on matrix multiplication using VSX and MMA instructions
https://www.redbooks.ibm.com/redpapers/pdfs/redp5612.pdf

D.1 Getting started with IBM POWER10 functional simulator

a. Write a simple C program that prints hello world". Run this program in POWER10 functional simulator and upload the screenshot of the output.

D.2 VSX extensions for matrix multiplication

a. Using the matmul.c as a starting code (Exercise→ Code → CH1 → matmul.c), vectorize the matrix multiplication. (You may use the sgemm VSX kernel from https://www.redbooks.ibm.com/redpapers/pdfs/redp5612.pdf)
b. Compare and analyze the performance of manually vectorized matmul.c with the baseline matmul.c.
c. For the manually vectorized code, compare the performance with all possible loop orders for matrix multiplication (ex. i-j-k, k-j-I, etc.). Is there a loop order that performs the best? If so, qualitatively explain why

---

D.3 MMA extensions for matrix multiplication

a. Using the matmul.c as a starting code (Exercise→ Code → CH1 → matmul.c), vectorize the matrix multiplication. (You may use the *SGEMM kernel using MMA instructions* from https://www.redbooks.ibm.com/redpapers/pdfs/redp5612.pdf)
b. How does the performance of matrix multiplication with MMA compare against VSX matrix multiplication? Qualitatively discuss the difference in the performance, if any.

D.4 (Optional)

a. Using the matmul.c as a starting code (Exercise→ Code → CH1 → matmul.c), vectorize the matrix multiplication using any one of the advanced optimizations (ex. multiple accumulators, cache-blocking, etc.) discussed in the chapter 4 of MMA best practices guide.
b. How does the performance of matrix multiplication with additional optimizations compare against matrix multiplication with MMA and VSX? Qualitatively discuss the difference in the performance, if any.