Assignment –

* Go to Matrix Multiplication code on Godbolt and run the c code with different compilers like RISC-V , X86 etc and note down your observations
* Change values/code in the c code and note down the changes in the assembly take multiple screen shots

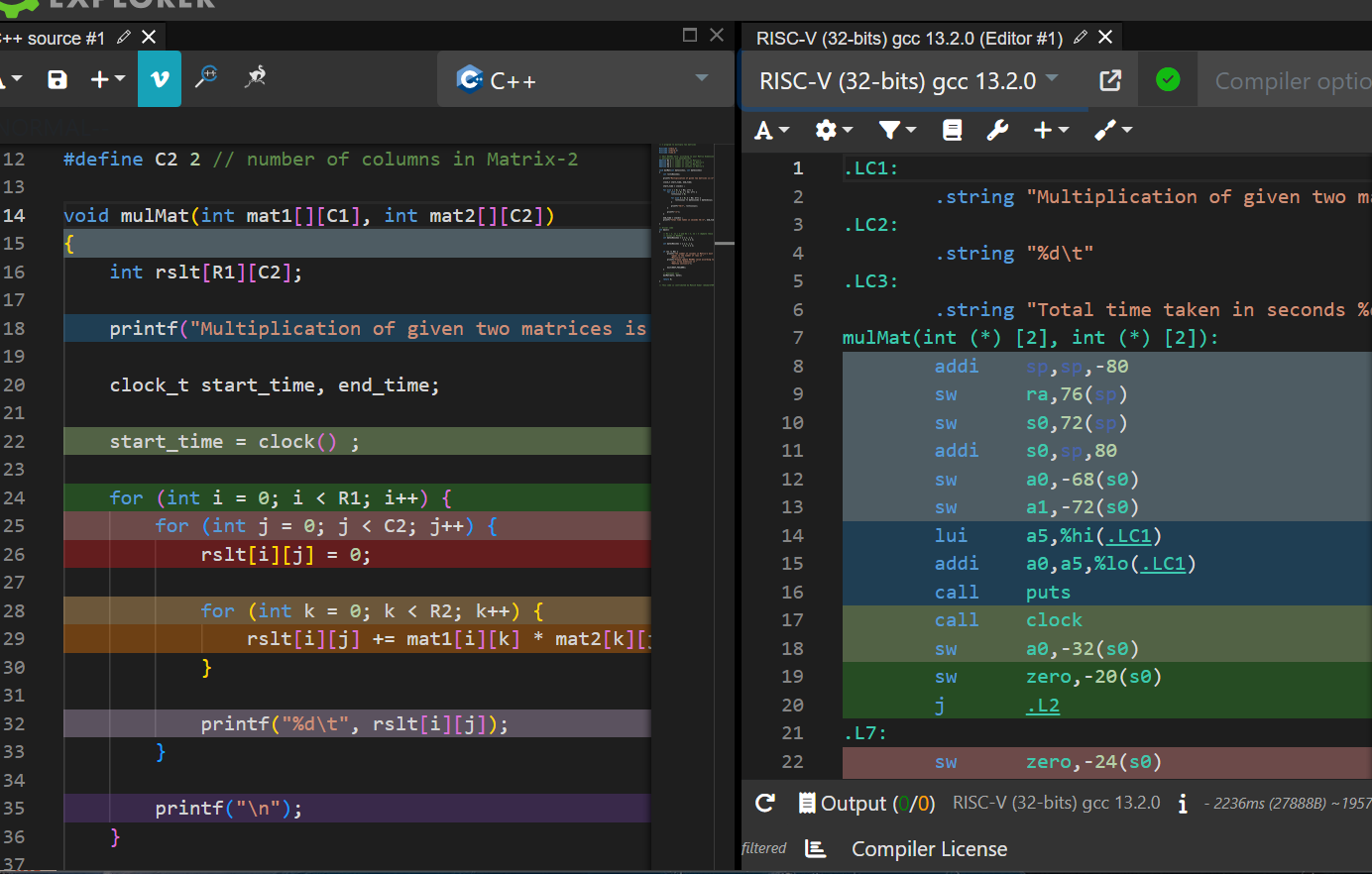
2x2 Matrix multiplication –

Observations about the C++ Code–

1. C++ Program initializaes 2 2x2 matrices and calls mulMat function to multiply them and print the result including the time taken to perform the calculation
2. Function mulMat uses multi dim arrays, time variables and nested for loops
3. Run the Program with RISC-V (32-bit) gcc13.2.0

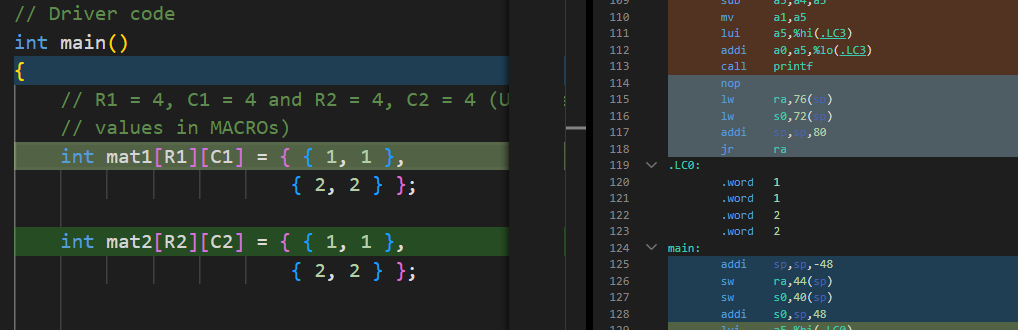
Observations about the assembly code –

1. The compiler compiles the program in the same order in which it’s written in C++ code
2. Compiler break downs the program in segments or blocks like .LC1:, .LC2:
3. One segment for each for loop
4. Each of these segments contains the assembly version of the c++ code which are the sequences of load, store ,mathematical operations, function calls and branch instruction
5. Total 159 lines of assembly code

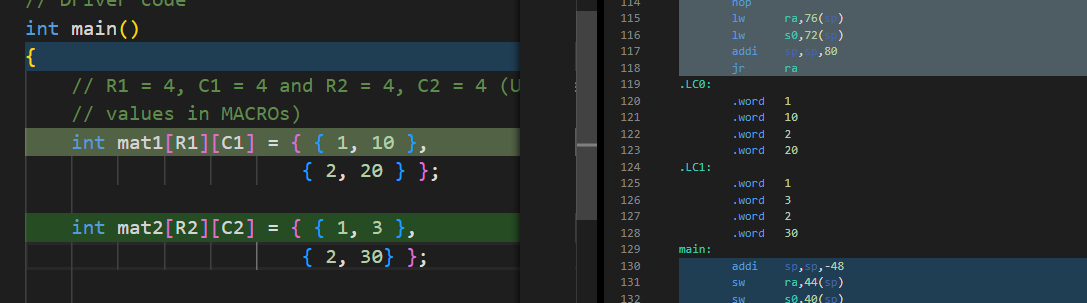


Observation after changing the numeric values in the input matrix –

1. Before changing the input



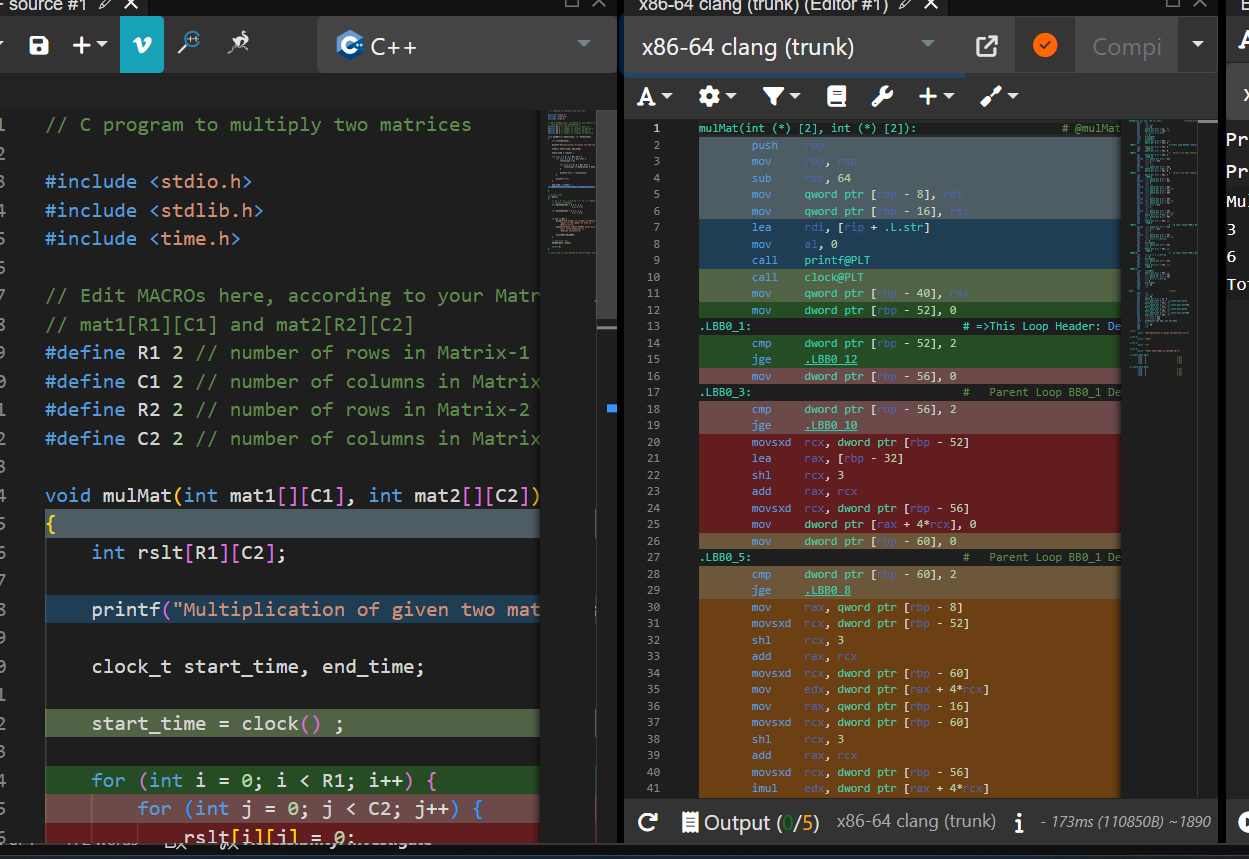
1. After Changing the input , changes are reflected in the assembly code . Also the number of lines in the assembly code increases from 149 to 164



**X86 , 64-bit compiler**

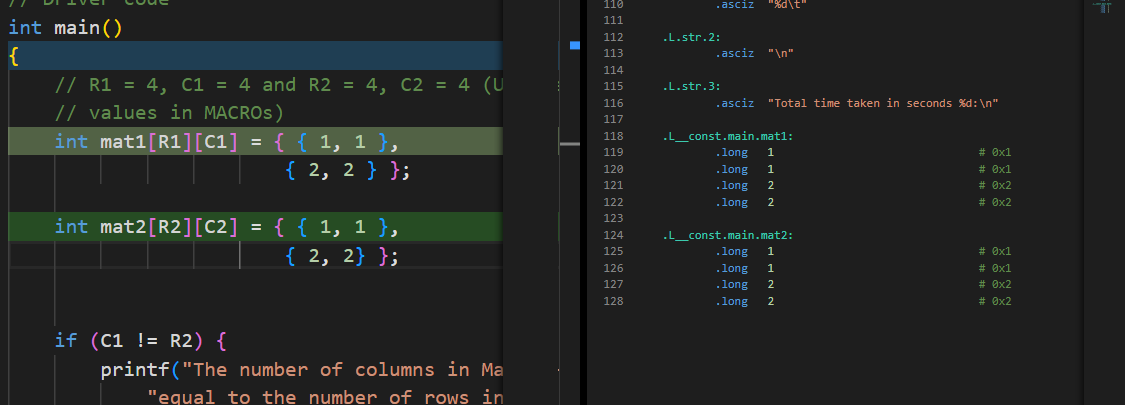
Assembly Code Observation-

1. More verbose and hence more understandable as compared to RISC-V assembly
2. Total 128 lines of code which is less than RISC-V assembly code
3. Similar observation about segments as RISC-V assembly code



1. Observation of assembly code on changing the numeric values in the input matrices

Before change –



After change observations –

1. Changes are reflected in the assembly code , however unlike RISC-V assembly code , the total number of lines in the assembly code remain the same to 128 lines.

