Assignment –

* Go to Counter on Godbolt and run the c code with different compilers like RISC-V , X86, Adrino, ARM 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

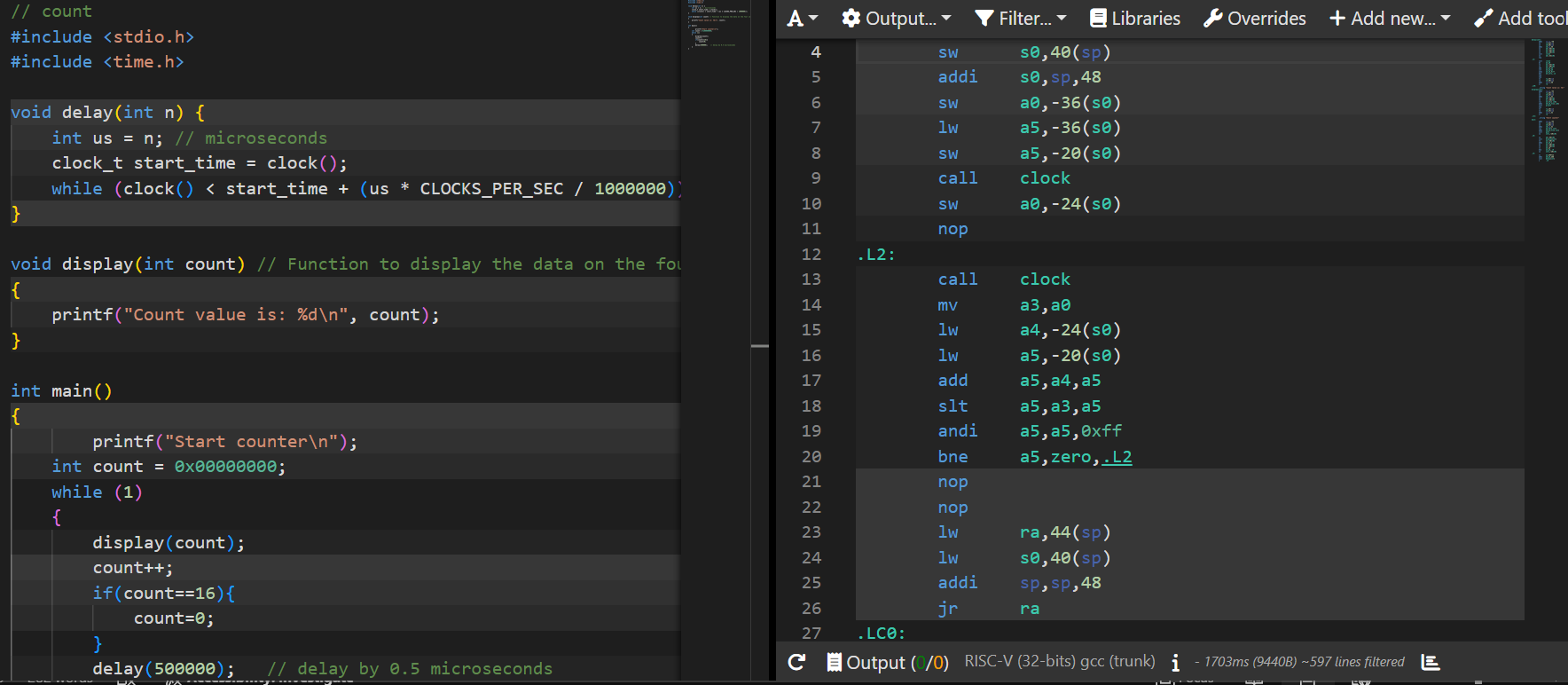
Counter–

Observations about the C Code–

1. The program increments a counter every 0.5 microsecond and resets the counter when the count reaches 16
2. Function uses clock , while loops, multiplication and division and comparison operations and print strings

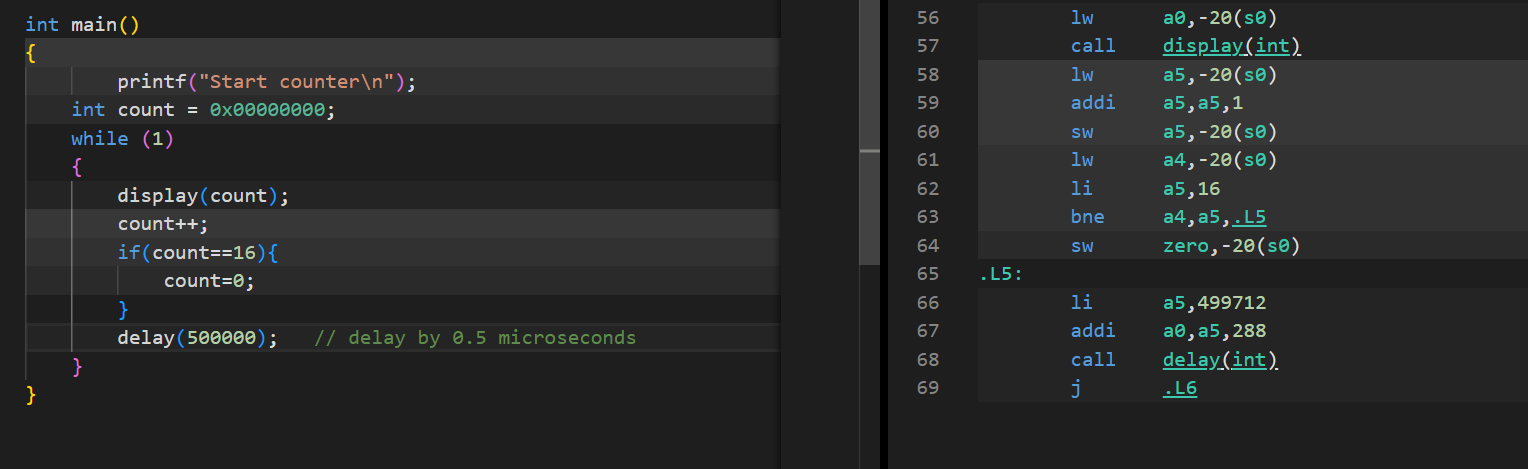
Observations about the assembly code –

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 .L1: L2:
3. One segment for each function or 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 69 lines of assembly code

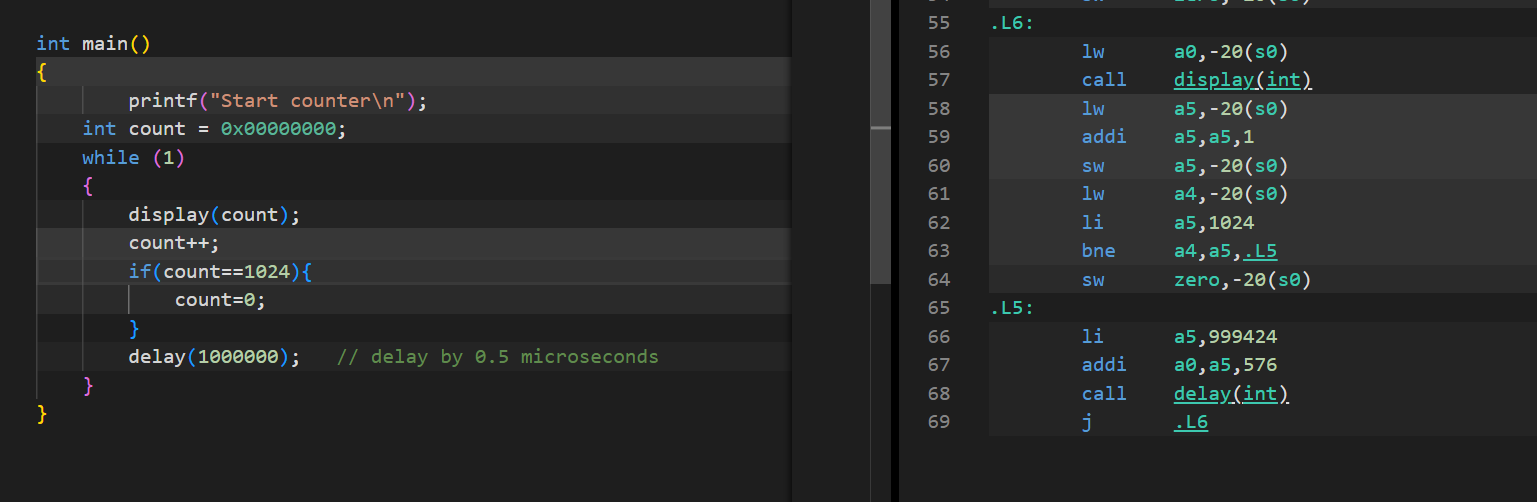


Observation after changing the numeric values in the input matrix –

1. Before changing the input



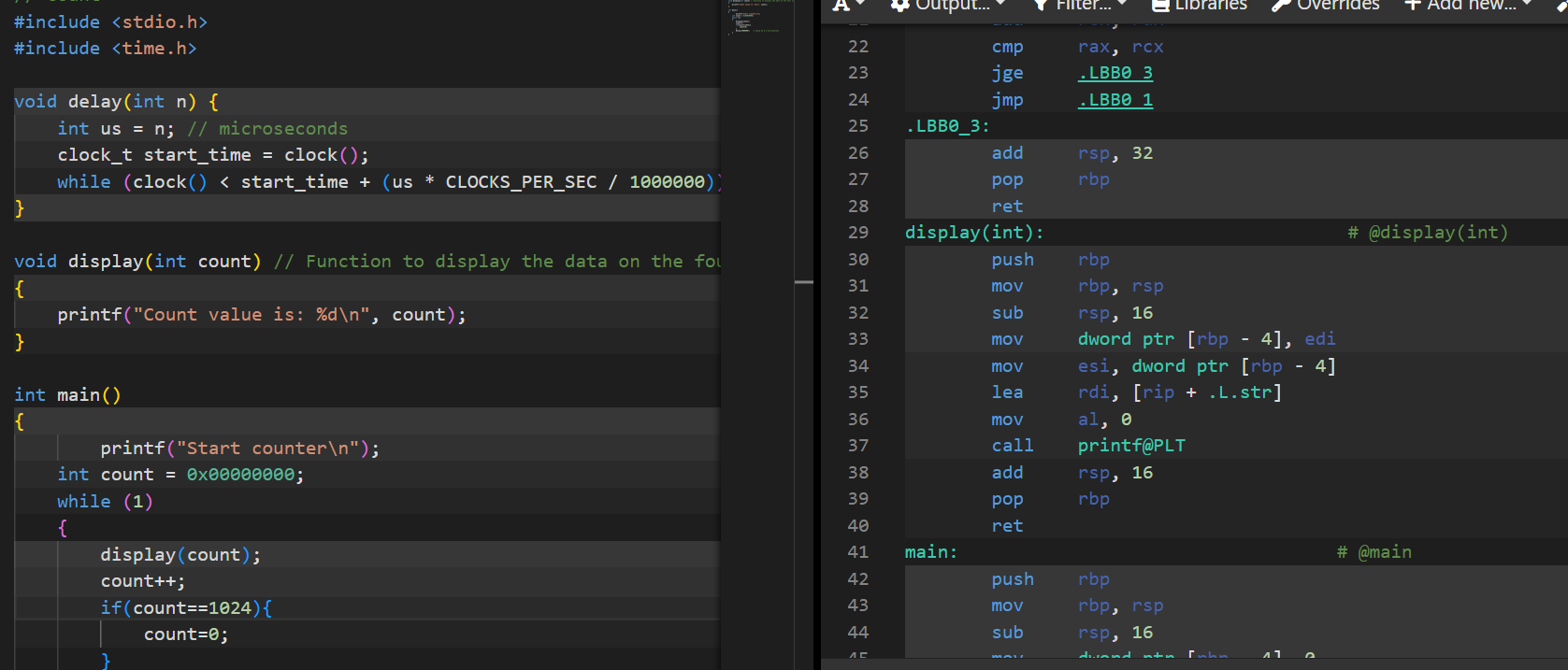
1. After Changing the input , changes are reflected in the assembly code .



**X86 , 64-bit compiler**

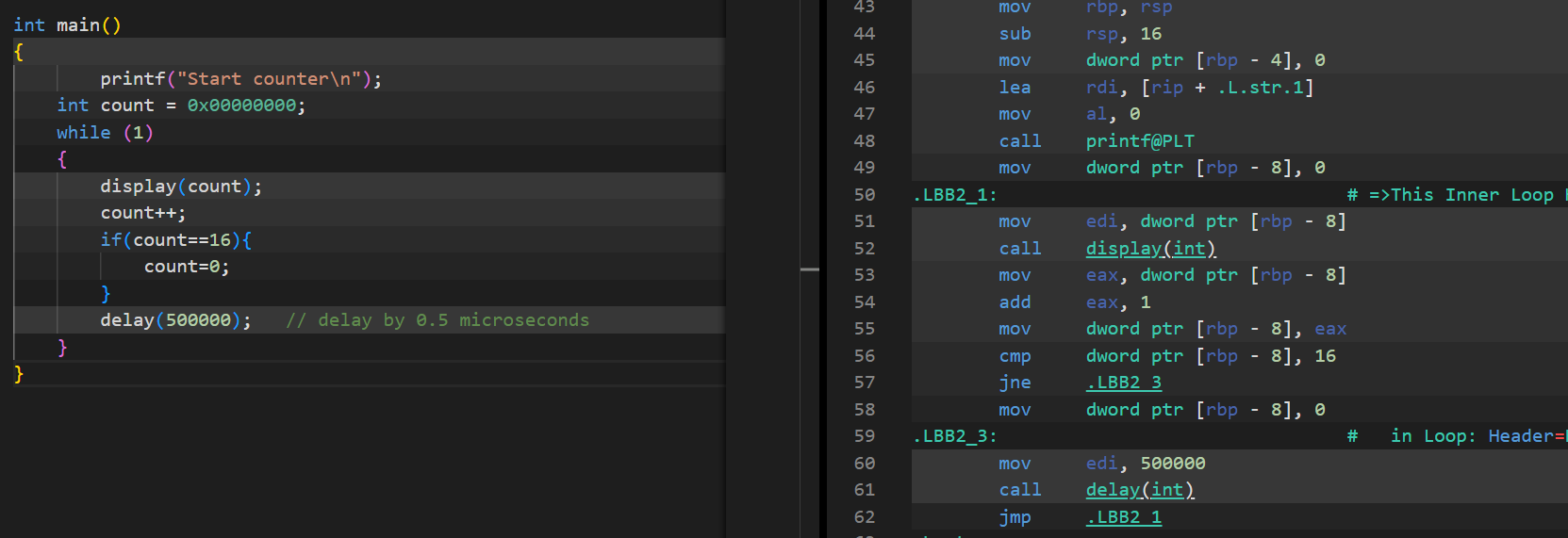
Assembly Code Observation-

1. More verbose and hence more understandable as compared to RISC-V assembly
2. Total 67 lines of code which is almost same to 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 counter

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 67 lines.

