**CSE Practices Lab 3**

Part A

The **Little Man Computer** (**LMC**) is an instructional [model](https://en.wikipedia.org/wiki/Model_(abstract)) of a [computer](https://en.wikipedia.org/wiki/Computer), created by Dr. [Stuart Madnick](https://en.wikipedia.org/wiki/Stuart_Madnick) in 1965.[[1]](https://en.wikipedia.org/wiki/Little_man_computer#cite_note-1) The LMC is generally used to teach students, because it models a simple [von Neumann architecture](https://en.wikipedia.org/wiki/Von_Neumann_architecture) computer—which has all of the basic features of a modern computer. It can be programmed in machine code (albeit in decimal rather than binary) or assembly code.[[2]](https://en.wikipedia.org/wiki/Little_man_computer#cite_note-2)[[3]](https://en.wikipedia.org/wiki/Little_man_computer#cite_note-3)[[4]](https://en.wikipedia.org/wiki/Little_man_computer#cite_note-4)

The LMC model is based on the concept of a little man shut in a closed mail room (analogous to a computer in this scenario). At one end of the room, there are 100 mailboxes ([memory](https://en.wikipedia.org/wiki/Memory_(computers))), numbered 0 to 99, that can each contain a 3 digit instruction or data (ranging from 000 to 999). Furthermore, there are two mailboxes at the other end labeled **INBOX** and **OUTBOX** which are used for receiving and outputting data. In the center of the room, there is a work area containing a simple two function (addition and subtraction) calculator known as the [Accumulator](https://en.wikipedia.org/wiki/Accumulator_(computing)) and a resettable counter known as the Program Counter. The Program Counter holds the address of the next instruction the Little Man will carry out. This Program Counter is normally incremented by 1 after each instruction is executed, allowing the Little Man to work through a program sequentially. [Branch](https://en.wikipedia.org/wiki/Branch_(computer_science)) instructions allow iteration (loops) and [conditional](https://en.wikipedia.org/wiki/Conditional_(programming)) programming structures to be incorporated into a program. The latter is achieved by setting the Program Counter to a non-sequential memory address if a particular condition is met (typically the value stored in the accumulator being zero or positive).

<https://en.wikipedia.org/wiki/Little_man_computer>

https://www.futurelearn.com/info/courses/how-computers-work/0/steps/49285

*You can find common Instructions*

<https://peterhigginson.co.uk/lmc/>

**Hint Program : Add two numbers**

INP

STA 99

INP

ADD 99

OUT

HLT

**Hint Program : Countdown on an Input Number**

INP

OUT // Initialize output

LOOP BRZ QUIT // Label this memory address as LOOP. If the accumulator value is 0, jump to the memory address labeled QUIT

SUB ONE // Subtract the value stored at address ONE from the accumulator

OUT

BRA LOOP // Jump (unconditionally) to the memory address labeled LOOP

QUIT HLT // Label this memory address as QUIT

ONE DAT 1 // Store the value 1 in this memory address, and label it ONE (variable declaration)

Problem 1.

Add three number and display result at output

Problem 2.

Subtract a number from another and display result at output

Problem 3.

Can you square an input number and display result at output

Problem 4.

Can you Generate Fibonacci Series (1st 5 terms)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part B

<https://onecompiler.com/assembly>

**Compiler Explorer** is an interactive online compiler which shows the assembly output of compiled C++, Rust, Go (and many more) code.

<https://godbolt.org/>

Problem 5,6,7 : Check Hello World, Squaring number, and Addition of two number programs in 5 variations