

Little Man Computer (LMC)

For this lab task you will be creating programs using the assembly language simulator Little Man Computer (LMC) through: <http://peterhigginson.co.uk/lmc/>
Each Exercise has an example piece of code which does some *similar* task.

Exercise 1: Create a program which takes in three numbers and stores them. The program should then output the sum of the first two numbers, with the third subtracted.

```

                INP
                STA     NUMONE
                OUT
                HLT
NUMONE         DAT
```

Exercise 2: Create a program which allows the user to input numbers indefinitely and outputs the running total after each entry. the program should stop when the number 0 is entered.

```

START          INP
                ADD     TOTAL
                OUT
                BRA     START
                HLT
TOTAL          DAT     000
```

Exercise 3: Create a program which takes in two numbers and outputs the answer of the two inputs multiplied together.

```

                INP
LOOP           STA     NUMONE
                SUB     ONE
                STA     NUMONE
                BRZ     END
                BRA     LOOP
                HLT
END            OUT
                HLT
NUMONE         DAT
ONE            DAT     001
```

Syntax Cheat Sheet

---	INP	---
---	OUT	---
---	HLT	---
---	STA	var
---	LDA	var
var	DAT	---
---	ADD	var
---	SUB	var
---	BRA	var
---	BRZ	var
---	BRP	var

Example Program

Create a program which takes in one input and outputs the positive value, i.e. if it is a negative, you should output the positive, e.g. inputting -3 would output 3 otherwise just output the input.

	BRA	START
NUMONE	DAT	
POSNUM	DAT	000
ONE	DAT	001
START	INP	
	BRP	OUTIN
	STA	NUMONE
MAKEZERO	LDA	POSNUM
	ADD	ONE
	STA	POSNUM
	LDA	NUMONE
	ADD	ONE
	STA	NUMONE
	BRZ	OUTPOS
	BRA	MAKEZERO
OUTIN	OUT	
	HLT	
OUTPOS	LDA	POSNUM
	OUT	
	HLT	

- Our first block of code branches over itself (we do not want the computer to run our data)! The rest of the block is reserving memory locations for our data.
- Our second block of code, which starts with the named memory location 'START', takes in an input and if it is zero or positive branches to 'OUTIN', if not stores the input into 'NUMONE' and starts 'MAKEZERO'.
- 'MAKEZERO' adds one to 'POSNUM', then adds one to 'NUMONE', if the result of adding one to 'NUMONE' gives zero then branch to 'OUTPOS' otherwise it branches back to 'MAKEZERO'.