



A9.5 Assembly language: the Little Man Computer

Computer architecture · A level · OCR H446 1.2.4, AQA 7517 4.7.3.5 · about 25 min

Name _____

Class _____

Date _____

What this lesson is about

OCR's LMC instruction set: tracing and writing programs with selection and iteration, run in Python.

- The model
- The instruction set
- A first program: add two numbers
- Selection: which is bigger?
- Iteration: a countdown
- Running LMC in Python
- How to write LMC in an exam

Questions

5 marks in all

1. In the LMC, what does BRP do? [1 mark]
☐ A Branches if the accumulator is zero or positive
☐ B Branches if the accumulator is positive only
☐ C Branches always
☐ D Branches if the accumulator is zero
2. Which LMC mnemonic stores the accumulator in a mailbox? [1 mark]

3. An LMC program is INP, STA 9, INP, SUB 9, OUT, HLT. The inputs are 20 then 35. What is output? [1 mark]

4. Why are DAT lines put after HLT? [1 mark]
☐ A So the little man never fetches data and tries to execute it as an instruction
☐ B Because the assembler only reads DAT at the end
☐ C To make the program run faster
☐ D Because data must be in mailbox 99

5. These mailboxes hold an assembled LMC program, run by this code. What does it output?

[1 mark]

```
mailboxes = [901, 308, 508, 902, 209, 308, 802, 0, 0, 1]
acc, pc, inputs = 0, 0, [2]
while True:
    ins = mailboxes[pc]
    pc += 1
    op, xx = ins // 100, ins % 100
    if ins == 0: break
    elif op == 2: acc -= mailboxes[xx]
    elif op == 3: mailboxes[xx] = acc
    elif op == 5: acc = mailboxes[xx]
    elif op == 8 and acc >= 0: pc = xx
    elif ins == 901: acc = inputs.pop(0)
    elif ins == 902: print(acc)
```

The task: the larger of each pair

Write an LMC program in the string `program`. It repeatedly inputs a pair of numbers, first and second, and outputs the larger of the two (either one if they are equal). If the first number of a pair is 0, it stops at once without reading a second number. The inputs are whole numbers from 0 to 999. The task's inputs are 58, 23, 7, 19, 30, 30, 0, so the program must output `OUT 58`, `OUT 19` and `OUT 30`, and nothing else. The folded helpers `assemble` and `run_lmc` are given: `run_lmc` prints each output as `OUT` and the value, and reads each input with `input()`. The robot does not move.

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/a9-5-the-little-man-computer/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Write an LMC program that inputs two numbers and outputs their product, by repeated addition.
2. Change the countdown so it stops at 1 instead of 0.
3. What would happen if the `DAT` lines were put at the start of the program instead of the end?