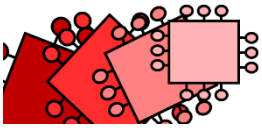
Computer Science A Level

Learners must take three components (01, 02 and 03 or 01, 02 and 04) to be awarded the OCR A Level in Computer Science.

Content Overview	Assessment Overview	
• The characteristics of contemporary processors, input, output and storage devices • Software and software development • Exchanging data • Data types, data structures and algorithms • Legal, moral, cultural and ethical issues • Elements of computational thinking • Problem solving and programming • Algorithms to solve problems and standard algorithms *The learner will choose a computing problem to work through according to the guidance in the specification.* • Analysis of the problem • Design of the solution • Developing the solution • Evaluation	Computer systems (01) 140 marks 2 hours and 30 minutes written paper (no calculators allowed)	**40%** of total A level
	Algorithms and programming (02*) 140 marks 2 hours and 30 minutes written paper (no calculators allowed)	**40%** of total A level
	Programming project 03* – Repository or 04* – Postal or 80 – Carry forward (2018 onwards)* 70 marks Non-exam assessment	**20%** of total A level



Computer Science A Level

CPU Architecture –

Use the below links for support –

https://www.youtube.com/watch?v=UdHK35N-Kuo&list=PLCiOXwirraUB7V2i0SJ4SSJFqRV_LtgzW

https://www.teach-ict.com/2016/A_Level_Computing/OCR_H446/1_1_characteristics_components/111_architecture/parts_of_cpu/miniweb/index.php

Teach ICT log in –

U: b913nz

P: school

Using the above complete the **CPU Theory Notes** and **CPU Exam Question**.

Little Man Computer –

Use the below link for support

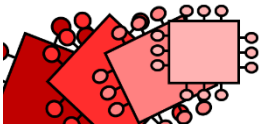
https://www.teach-ict.com/2016/A_Level_Computing/OCR_H446/1_2_software/124_assembly/lmc/miniweb/index.php

and attempt the below tasks

https://www.teach-ict.com/xml/submainlogin.php?fn=/2016/A_Level_Computing/OCR_H446/1_2_software/124_assembly/lmc/lmc_tasks/tasks.php

Use the **How to use LMC.docx** for help

Then attempt the tasks **Assembly Language Activity.docx**



Theory Notes Task: The CPU

1. State the main role of the CPU

2. The CPU also controls other physical hardware within the computer. List three examples of hardware that it might control.

a.

b.

c.

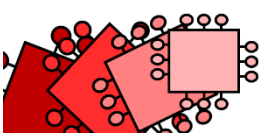
3. State where the CPU is located within a computer

4. List the three main parts found within a CPU

1.

2.

3.



5. What is used to carry data within the CPU and also to and from main memory?

6. Identify the three tasks that the Control Unit performs

a.

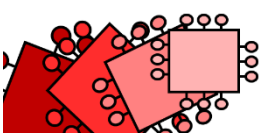
b.

c.

7. Identify the two tasks that the ALU performs

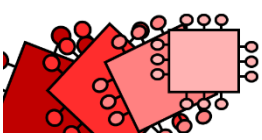
a.

b.



8. Briefly explain the role of a register within the CPU
9. State the purpose of the cache

10. Explain why there is a clock inside the CPU

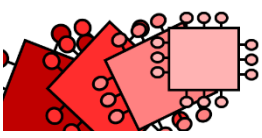


Exam Style Questions: The CPU

1a. The CPU is where the main storage occurs	True	False (1)
1b. The CPU fetches instructions from memory	True	False (1)
1c. The CPU encodes the instructions from a program	True	False (1)
1d. The CPU executes instructions in order to process data	True	False (1)

2. Describe how the CPU and RAM work together to enable data to be processed (2 marks)

3. Explain the purpose of cache memory (3 marks)



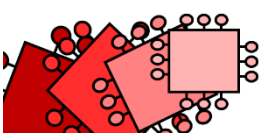
4. Identify two of the roles performed by the Control Unit (2 marks)

5. The Arithmetic Logic Unit (ALU) performs two main tasks, the arithmetic operation and the logic operation.

a. Outline what happens during the arithmetic operation (1 mark)

b. Outline what happens during the logic operation (1 mark)

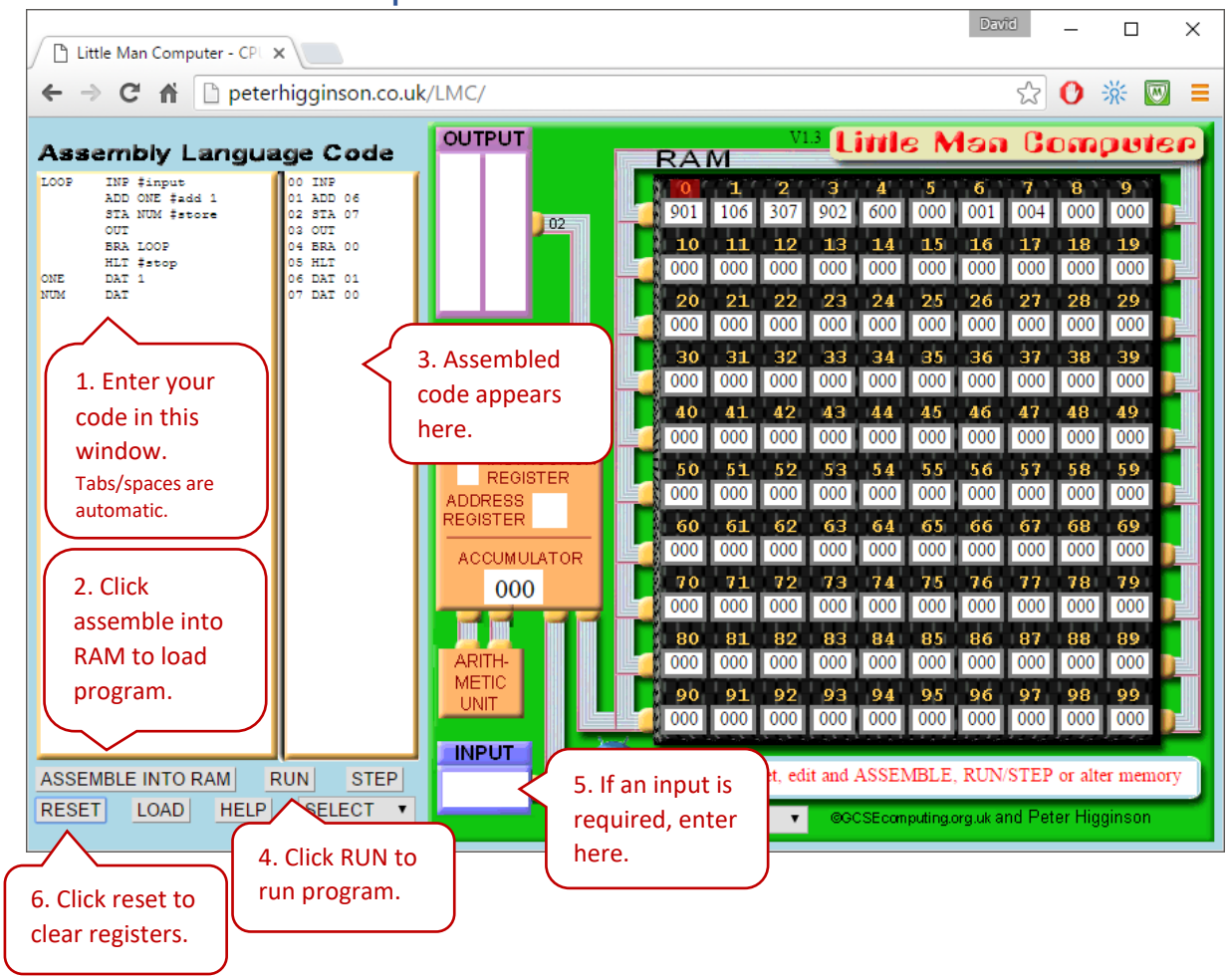
6. Explain the purpose of a register within the CPU (2 marks)



Where to find Little Man Computer

http://peterhigginson.co.uk/LMC/

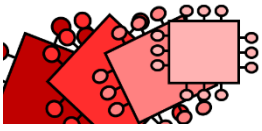
The Little Man Computer interface



Getting started in Assembly language

Enter this program into the code window. It returns true (-1) if the number entered is positive and false (0) if the number input is negative. The notes on the right explain the program. You do not need to enter those.

```
INP                 # Waits for a number to be input.
STA number          # Stores the number in memory.  Note 'number' is a
variable.
BRP positive        # Branches to the label 'positive' if the number is positive.
LDA false           # Loads the value for false (0) into the accumulator.
OUT                 # Outputs the accumulator.
HLT                 # Stops the program.
positive LDA true   # Loads the value for true (-1) into the accumulator.
OUT                 # Outputs the accumulator.
HLT                 # Stops the program.
number DAT          # Reserves a memory location to store the number.
false DAT 0         # Reserves a memory location to hold 0.
true DAT -1         # Reserves a memory location to hold -1.
```



Running the program

Click 'Assemble into RAM'. This runs the translation assembler and puts the program in memory, changing the commands (mnemonics) into opcodes and operands.

Click 'Run'.

Note, there are no prompts in LMC. When a program needs an input it stops and waits for you to enter a number in the input window.

Enter the number 5 and see if -1 is output.

You can click 'Reset' to clear the registers ready for the program to be run again.

Points to note in this version of LMC

Loop entry points are indicated with labels before the command, e.g.

```
LOOP INP      (loop label and input command)
BRA LOOP      (branch to loop label)
```

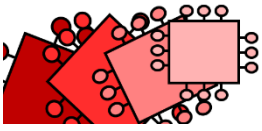
Comments can be entered with a # symbol at the end of a command.

Blank lines are supported to split up sections of the program.

DAT is used to declare a variable and can be used as a constant if given a value.

Commands

Mnemonic	**Instruction**
HLT	End program (Halt)
ADD *variable*	Add variable to accumulator
SUB *variable*	Subtract variable from accumulator
STA *variable*	Store accumulator to variable
LDA *variable*	Load variable to accumulator
BRA *label*	Branch always to label
BRZ *label*	Branch to label if accumulator is zero
BRP *label*	Branch to label if accumulator is greater than or equal to zero
INP	Input from keyboard to accumulator
OUT	Output from accumulator to display
variable DAT *value*	Declare variable



Introduction to programming

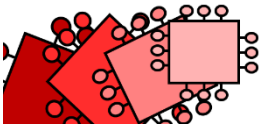
Assembly language (including following and writing programs with Little Man Computer)

Solve these problems in LMC and explain your solution

Write the following programs in LMC. Enter the assembly code instructions you used in the column titled, 'assembly language'. Explain what the command is doing in the explanation column so it makes sense to someone reading your program.

Input 3 numbers and output them in reverse order.	
Assembly language	Explanation

Input a number, add 5 to it and output the answer. 5 is not input at the keyboard, put it directly in memory.	
Assembly language	Explanation



Introduction to programming

Assembly language (including following and writing programs with Little Man Computer

Input 2 numbers. Subtract the first number from the second number and output the answer.	
Assembly language	Explanation

Input 2 numbers. Subtract the first from the second and output the answer, then subtract the second from the first and output the answer.	
Assembly language	Explanation