Practical 1

The purpose of this prac is to get you familiar with the writing, building and running code. It will encompass moving data in and out of the CPU and performing arithmetic operations.

Part 1:

On boot up, the micro should move the following words into RAM, beginning at the start of RAM. They are labeled here for future use in this document:

Label	Address (relative to the start of RAM)	Data
A	0	0xAABBCCDD
B	4	0x00001122
C	8	0x00002233
D	12	0x55555555

Once this copy to RAM operation is performed, the next line should be labeled:
copy_to_RAM_complete
This will be used by the automarker, so ensure it's correct.

Part 2:

Now that the data is in RAM, it must be read back from RAM so that operations are done on it. It is very important that you read the data in from RAM and not from Flash.
Perform operations to cause the following RAM addresses to take on the data.

Address	Computation	Result
16	A - B	0xAABBBBBB
20	A + B + C	0xAABC0032
24	B (EXCLUSIVE OR) C	0x00003311
28	B * C	0x0249EDC6

Bonus:

Display the contents of address 0x20000010 on the LEDs

Template for compliance with automarker:

```
@STDNUM001 STDNUM002

        .syntax unified
        .global _start

        .word 0x20001FFF
        .word _start + 1
_start:
        @copy data to RAM

copy_to_RAM_complete:
        @read data from RAM into registers and perform operations,
        @writing result back to RAM

infinite_loop:
        B infinite_loop

        .align
        @ whatever literals you need go here
```

My source file was 50 lines long. You should expect to produce similar. Any more and you're working too hard. Any less and you're making me look bad.

Marking:

- Part 1:
1 mark per correct word moved into RAM x 4 = 4

- Part 2:
1 marks per correct result x 4 = 4

- Timeline:
Before <day> at 10:00:
Monday: 3
Tuesday: 2
Wednesday: 1

- Bonus:
3 marks if correct

Marked out of: 11
Available marks: 14