Possible coding problems for test1

1. Load 0xAA into 10 consecutive memory locations starting at 0x40000000

2. Write assembly language instructions to add values stored at 10 consecutive locations in memory (assume a label "xdata" is available that defines the start of the x array). Create functionality similar to the for loop below:

```
y = 0;
for (i = 1; i<=10; i++)
{
    y = y + x(i)
}
```

3. Write assembly language instructions to move a word of data from location ydata1, preserve only bits 14-17, move those bits to the highest-order four bits, and then store the result at ydata2. Set the ALU flags as a result of the move.

4. Write an assembly language program to add the number stored at "dataloc1" to the number stored at "dataloc2" and store the result in dataloc2, but only if the 4^{th} bit of dataloc1 is a '1'. If the 4^{th} bit is a '0', set both dataloc1 and dataloc2 to all 0's. When complete, drop into the infinite loop.

5. Write an assembly language program to put the largest of 8 numbers stored in consecutive memory locations starting at "dataset1" into R1.

```
.text
.global main
.equ dataset1, 0x1000

LDR R0, =dataset1
```

6. Using load and store instructions, move a block of 24 words from memory area "block1" to memory area "block2", and then fall into the infinite loop. Use as many GPRs as you would like.

7. Write an assembly program to load 40 consecutive data words starting from "list1", add them to 40 consecutive data words starting from "list2", and store the results in 40 consecutive locations staring from "results1". Also, add all 80 data words together into R9. When all 40 locations have been added, enter the infinite loop.

8. Write assembly instructions to store R4 – R14 into ascending memory starting at a memory location identified by the label "stack". The fewer instructions you use, the more points you get (provided your solution works).

```
.text
.global main
.equ stack, 0x01010000
```