```
/* Program that counts consecutive 1's */

            .text                   // executable code follows
            .global _start
_start:
            MOV     R1, #TEST_NUM   // load the data word ...
            LDR     R1, [R1]        // into R1

            MOV     R0, #0          // R0 will hold the result
LOOP:       CMP     R1, #0          // loop until the data contains no more 1's
            BEQ     END
example     LSR     R2, R1, #1      // perform SHIFT, followed by AND
            AND     R1, R1, R2
            ADD     R0, #1          // count the string length so far
            B       LOOP

END:        B       END

TEST_NUM:   .word   0x103fe00f

            .end
```

Illustrate with a different num: 0x18F3

orig: R1 : 0001 1000 1111 0011 R0

LSR R1,R2: 0000 1100 0111 1001 0

AND R1,R1,R2: 0000 1000 0111 0001

LSR : 0000 0100 0011 1000 1

AND : 0000 0000 0011 0000

LSR: 0000 0000 0001 1000 2

AND: 0000 0000 0001 0000

LSR: 0000 0000 0000 1000 3

AND : 0000 0000 0000 0000 4 (after AND) → end LOOP

Memory Mapped Input/Output

Recall: memory is the idea that "special" memory addresses are used to transmit data from/to input/output devices

e.g. on DE1-SoC:

Address 0xFF20 0000 LEDR } STR
" 0010 HEx 5:0 }
" 0040 SW 9:0 } LDR
" 0050 KEY 3:0 }

e.g. copy switches into LEDS

```
        MOV   R0, #ID_ADDR
        LDR   R0, [R0]          // R0 ← FF20 0000  LEDR
MAIN:   LDR   R1, [R0, #0x40]   // R1 ← SW
        STR   R1, [R0]          // LED ← SW
        B     MAIN
```

Part 4:

must count 1's, 0's, 0101···3 display two digit decimal numbers on HEX display

e.g: suppose you have 16 1's in a row

@ address FF20 0020