```
;***************************************************************************
;*
;* Title:           segment_and_digit_test
;* Author:          Jason Chen
;* Version:         1
;* Last updated:    10/12/2022 12:11:00
;* Target:          AVR128DB48
;*
;* DESCRIPTION
;*  Task 1
;*  Continually display the digit 8 and decimal point for approximately
;*  one second at each digit position from right to left.
;*
;* VERSION HISTORY
;* 1.0 Original version
;***************************************************************************

start:
    ldi r16, 0xFF       ; load r16 with all 1s
    out VPORTD_DIR, r16 ; VPORTD - all pins configured as outputs
    out VPORTA_DIR, r16 ; VPORTA - all pins configured as outputs
    ldi r16, 0x00       ; load r16 with all 0s
    out VPORTC_DIR, r16 ; VPORTC - all pins configured as inputs
    cbi VPORTE_DIR, 0   ; set direction for PE0 as input
    sbi VPORTE_DIR, 1   ; set direction for PE1 as output
    out VPORTD_OUT, r16 ; a-g and dp ON

main_loop:
    ldi r16, 0xE0
    out VPORTA_OUT, r16 ; digit 4 (rightmost) ON
    rcall one_sec_delay
    ldi r16, 0xD0
    out VPORTA_OUT, r16 ; digit 3 ON
    rcall one_sec_delay
    ldi r16, 0xB0
    out VPORTA_OUT, r16 ; digit 2 ON
    rcall one_sec_delay
    ldi r16, 0x70
    out VPORTA_OUT, r16 ; digit 1 (leftmost) ON
    rcall one_sec_delay
    rjmp main_loop

;***************************************************************************
;*
;* "one_sec_delay" - One Second Delay
;*
;* Description:     Two registers are subtracted from 5202 to 0, taking
;*                  1 second to execute.
;* Author:          Professor Ken Short
;* Version:         1
;* Last updated:    10/13/2022
;* Target:          AVR128DB48
;* Number of words:
;* Number of cycles:
;* Low registers modified:  n/a
;* High registers modified: r30, r31
;*
;* Parameters:  n/a
;*
;* Returns:     n/a
;*
;* Notes:       n/a
;*
;***************************************************************************

; 1.00008575 seconds @ 4 MHz system clock, 192 us resolution
one_sec_delay:
    ldi r30, LOW(5202)  ; outer loop 16- bit iteration count
    ldi r31, HIGH(5202) ; 16-bit value in r31:r30
    outer_loop:
        ldi r18, $FF    ; inner loop 8-bit iteration count
    inner_loop:
        dec r18         ; subtract 1 from inner loop count
        brne inner_loop
        sbiw r31:r30, 1 ; subtract 1 from outer loop count
        brne outer_loop
    ret
```