; Assembler equates  
PORTT = $00AE ; input port **for** DELAY\_CNT  
DDRT = $00AF  
PORTJ = $0028 ; output port **for** LEDs  
DDRJ = $0029  
LED\_MSK = 0b01100000 ; LED output pins  
R\_LED = 0b00100000 ; red LED output pin  
G\_LED = 0b01000000 ; green LED output pin  
  
  
; RAM area  
.area bss  
DELAY\_CNT:: .blkb 1  
  
; code area  
.area text  
;  
;===============================================================  
;  
; main  
\_main::  
  
 jsr SETUP ; jump to SETUP subroutine  
  
; YOUR CODE GOES HERE  
  
 bset PORTJ, *"Either R\_LED, G\_LED, or MASK"* ; Green ON and red OFF  
 bclr PORTJ,   
 jsr DELAY;  
 bclr PORTJ,   
 bclr PORTJ,   
 jsr DELAY;  
 bclr PORTJ,   
 bset PORTJ,   
 jsr DELAY;  
 bclr PORTJ,   
 bclr PORTJ,   
 jsr DELAY;  
 bset PORTJ,   
 bset PORTJ,   
 jsr DELAY;  
 bclr PORTJ,   
 bclr PORTJ,   
 jsr DELAY;  
   
   
; end main  
  
;===============================================================  
;  
; Subroutine Delay Delay[s] = ~100ms per DELAY\_CNT  
;  
DELAY:  
 ldaa PORTT ; (3) load 8-bit DELAY\_CNT  
 staa DELAY\_CNT ; (3)  
OUTER: ; outer loop  
  
 ldaa DELAY\_CNT ; (3)  
 cmpa #0 ; (1)  
 beq EXIT ; (1)  
 dec DELAY\_CNT ; (4)  
 ldy #$FFFC ; (2)  
INNER: ; inside loop  
 cpy #0 ; (2)  
 beq OUTER ; (1)  
 dey ; (1)  
 bra INNER ; (3)  
EXIT:  
 rts ; (5) exit DELAY  
  
; end subroutine DELAY  
;===============================================================  
;  
; Subroutine SETUP  
;  
SETUP:  
  
; setup IO ports  
  
 clr DDRT ; set PORTT to input  
 bclr PORTJ, LED\_MSK ; initialize LEDs to off  
 bset DDRJ,LED\_MSK ; set LED pins to output  
 rts ; exit SETUP  
  
;;;; end subroutine SETUP  
;===============================================================  
  
  
.area interrupt\_vectors (abs)  
 .org $FFFE ; at reset vector location,   
 .word \_\_start ; load starting address