```
;****************** main.s ***************
; Program written by: Dylan Cauwels
; Date Created: 1/20/2017
; Last Modified: 1/24/2017
; Brief description of the program
; The objective of this system is to implement a Car door signal system
; Hardware connections: Inputs are negative logic; output is positive logic
;  PF0 is right-door input sensor (1 means door is open, 0 means door is closed)
;  PF4 is left-door input sensor (1 means door is open, 0 means door is closed)
;  PF3 is Safe (Green) LED signal - ON when both doors are closed, otherwise OFF
;  PF1 is Unsafe (Red) LED signal - ON when either (or both) doors are open, otherwise OFF
; The specific operation of this system
;   Turn Unsafe LED signal ON if any or both doors are open, otherwise turn the Safe LED signal ON
;   Only one of the two LEDs must be ON at any time.
; NOTE: Do not use any conditional branches in your solution.
;       We want you to think of the solution in terms of logical and shift operations

GPIO_PORTF_DATA_R  EQU 0x400253FC
GPIO_PORTF_DIR_R   EQU 0x40025400
GPIO_PORTF_AFSEL_R EQU 0x40025420
GPIO_PORTF_PUR_R   EQU 0x40025510
GPIO_PORTF_DEN_R   EQU 0x4002551C
GPIO_PORTF_LOCK_R  EQU 0x40025520
GPIO_PORTF_CR_R    EQU 0x40025524
GPIO_PORTF_AMSEL_R EQU 0x40025528
GPIO_PORTF_PCTL_R  EQU 0x4002552C
GPIO_LOCK_KEY      EQU 0x4C4F434B  ; Unlocks the GPIO_CR register
SYSCTL_RCGCGPIO_R  EQU   0x400FE608
      THUMB
      AREA    DATA, ALIGN=2
;global variables go here
      ALIGN
      AREA    |.text|, CODE, READONLY, ALIGN=2
      EXPORT  Start

Start
    LDR R1, =SYSCTL_RCGCGPIO_R      ;Activate Port F Clock
    LDR R0, [R1]
    ORR R0, R0, #0x20
    STR R0, [R1]                    ;Clock Initialize Time
    NOP
    NOP
    LDR R1, =GPIO_PORTF_LOCK_R      ;Unlock PortF Register
    LDR R0, =0x4C4F434B
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_CR_R
    MOV R0, #0xFF
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_AMSEL_R     ;Disable Analog
    MOV R0, #0
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_PCTL_R      ;Start GPIO
    MOV R0, #0x00000000
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DIR_R       ;Set Direction Register
    MOV R0,#0x0A                    ;0 & 4 Input, 1-3 Output
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_AFSEL_R     ;Initialize PortF
    MOV R0, #0
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_PUR_R       ;Pull Up Resistors
    MOV R0, #0x11
    STR R0, [R1]
    LDR R1, =GPIO_PORTF_DEN_R       ;Enable PortF
    MOV R0, #0xFF
    STR R0, [R1]

loop                                ;PortF_Input
    LDR R1, =GPIO_PORTF_DATA_R  ;Load PortF Data Address
    LDR R0, [R1]                ;Load PortF Data
    AND R0,R0,#0x11             ;Isolate Input  Pins
    AND R1, R0, #0x01           ;Isolate SW1
```

```
        LSL R1, R1, #1              ;Move SW1 bit to common position
        AND R2, R0, #0x10           ;Isolate SW2
        LSR R2, R2, #3              ;Move SW2 bit to common position
        ORR R3, R2, R1              ;Check if either is on
        MOV R4, #2                  ;Move Result to LED
        EOR R5, R4, R3              ;Insert Result
        LSL R5, R5, #2              ;Move to other LED
        ADD R5, R5, R3              ;Insert Result
        LDR R1, =GPIO_PORTF_DATA_R  ;Reload Data Register Address
        STR R5, [R1]                ;Toggle LED's accordingly

        B   loop

          ALIGN        ; make sure the end of this section is aligned
          END          ; end of file
```