;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main.s \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Program written by: \*\*\*Elvin J. Galarza\*\*\*

; Date Created: 1/22/2016

; Last Modified: 9/6/2016

; Section \*\*\*Tuesday 3-4\*\*\*

; Instructor: \*\*\*Vijay Janapa Redi\*\*\*

; Lab number: 1

; Brief description of the program

; The overall objective of this system is a digital lock

; Hardware connections

; PE3 is switch input (1 means switch is not pressed, 0 means switch is pressed)

; PE4 is switch input (1 means switch is not pressed, 0 means switch is pressed)

; PE5 is switch input (1 means switch is not pressed, 0 means switch is pressed)

; PE2 is LED output (0 means door is locked, 1 means door is unlocked)

; The specific operation of this system is to

; unlock if all three switches are pressed

GPIO\_PORTE\_DATA\_R EQU 0x400243FC

GPIO\_PORTE\_DIR\_R EQU 0x40024400

GPIO\_PORTE\_AFSEL\_R EQU 0x40024420

GPIO\_PORTE\_DEN\_R EQU 0x4002451C

GPIO\_PORTE\_AMSEL\_R EQU 0x40024528

GPIO\_PORTE\_PCTL\_R EQU 0x4002452C

SYSCTL\_RCGCGPIO\_R EQU 0x400FE608

AREA |.text|, CODE, READONLY, ALIGN=2

THUMB

EXPORT Start

Start

PortE\_Init ;// Steps for initiating Port E

LDR R1, =SYSCTL\_RCGCGPIO\_R ;// Step 1) following lines activate clock for Port E (this line gets the address of SYSCTL\_RCGCGPIO\_R and puts it into R1)

LDR R0, [R1] ;// gets the data stored at the address of SYSCTL\_RCGCGPIO\_R and puts it into R0

ORR R0, R0, #0x10 ;// sets bit 4 of SYSCTL\_RCGCGPIO\_R's data to turn on Port E's clock

STR R0, [R1] ;// puts modified data back into R1, indicating we only want to deal with Port E

NOP

NOP ;// allows time for clock to finish

LDR R1, =GPIO\_PORTE\_DIR\_R ;// makes PE2 (LED) high

MOV R0, #0x04

STR R0, [R1]

LDR R1, =GPIO\_PORTE\_DEN\_R ;// enables Port E digital port (we're only working with the virtual "digital" port, not the physical "analog" one)

LDR R0, [R1]

MOV R0, #0x0FF

STR R0, [R1]

loop

;// Here is where the PE2 = (not(PE3)) and (not(PE4) and (not(PE5)) function is implemented

LDR R1, =GPIO\_PORTE\_DATA\_R ;// R1 is pointer for Port E

LDR R0, [R1] ;// R0 contains original input values

LSR R0, R0, #1 ;// shift bits over to the right, this allows PE3 to be at LED pin

MVN R3, R0 ;// flips bits R3 = NOT(PE3) in LED location

LDR R0, [R1]

LSR R0, R0, #2

MVN R4, R0 ;// R4 = NOT(PE4) in LED location

LDR R0, [R1]

LSR R0, R0, #3

MVN R5, R0 ;// R5 = NOT(PE5) in LED location

AND R6, R3, R4 ;// AND after NOT(PE3-5), where R6 =(NOT(PE3)) AND (NOT(PE4)) for line 64

AND R6, R5, R6

STR R6, [R1] ;// at this point, R6 will be holding the correct value for the LED pin and all this does is store that value into the DATA for PortE

B loop

ALIGN ; make sure the end of this section is aligned

END ; end of file