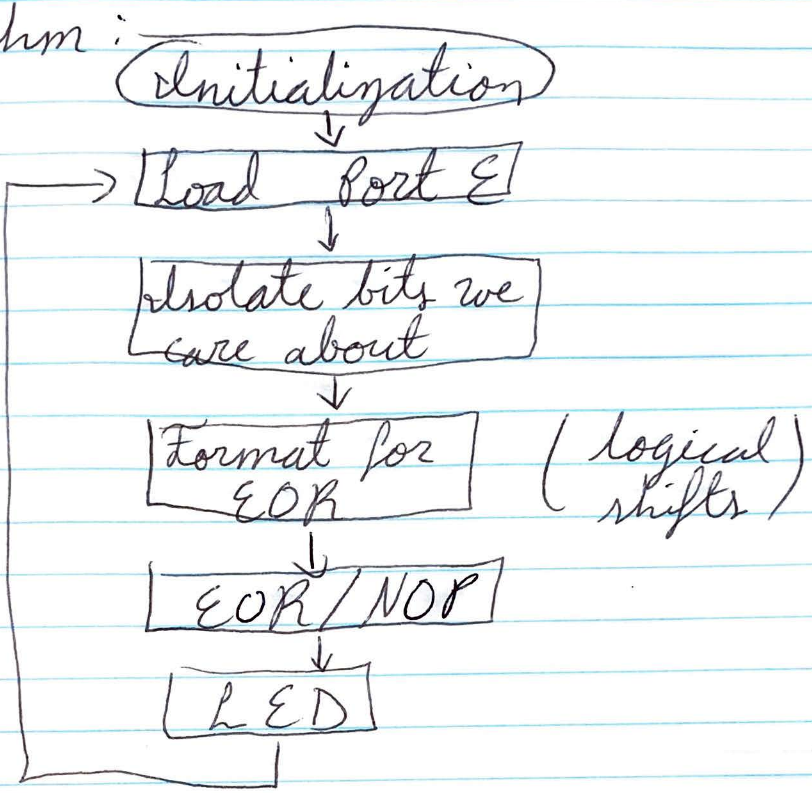
Algorithm:

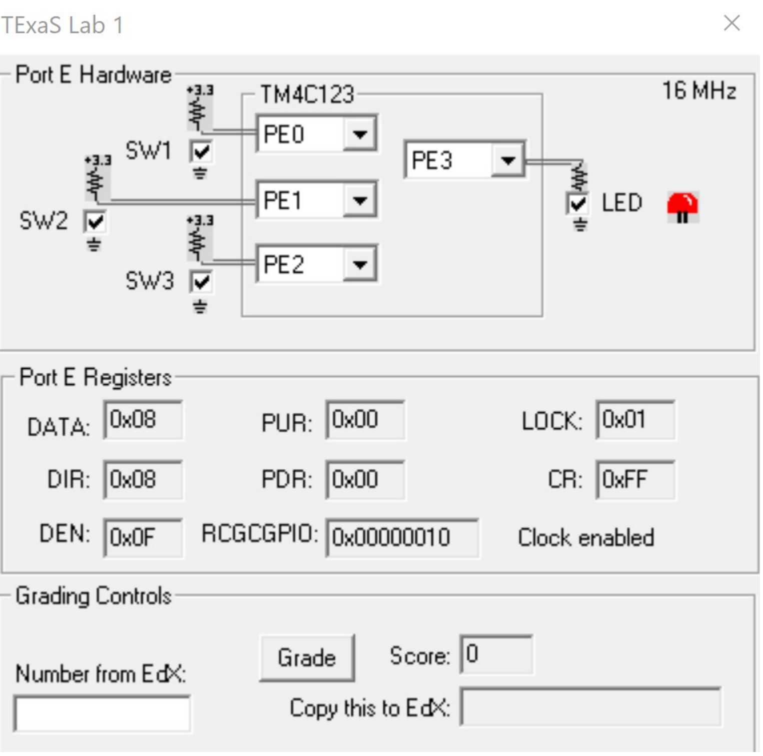


Pseudocode:

1. Turn on clock for for Port E
2. Set direction of Port E as output
3. Digital Enable pin 0 - pin 3
4. Read data from Port E
5. Isolate data for each corresponding input pin
6. Calculate inputs
7. Corresponding output should be displayed.

```
;****************** main.s ***************
; Program initially written by: Yerraballi and Valvano
; Author: Roberto Reyes
; Date Created: 1/15/2018
; Last Modified: 2/04/2020
; Brief description of the program: Solution to Lab1
; The objective of this system is to implement odd-bit counting system
; Hardware connections:
;  Output is positive logic, 1 turns on the LED, 0 turns off the LED
;  Inputs are negative logic, meaning switch not pressed is 1, pressed is 0
;    PE0 is an input
;    PE1 is an input
;    PE2 is an input
;    PE3 is the output
; Overall goal:
;   Make the output 1 if there is an odd number of switches pressed,
;     otherwise make the output 0

; The specific operation of this system
;   Initialize Port E to make PE0,PE1,PE2 inputs and PE3 an output
;   Over and over, read the inputs, calculate the result and set the output
; PE2  PE1 PE0  | PE3
; 0    0    0   |  1    odd number of 0’s
; 0    0    1   |  0    even number of 0’s
; 0    1    0   |  0    even number of 0’s
; 0    1    1   |  1    odd number of 0’s
; 1    0    0   |  0    even number of 0’s
; 1    0    1   |  1    odd number of 0’s
; 1    1    0   |  1    odd number of 0’s
; 1    1    1   |  0    even number of 0’s
;There are 8 valid output values for Port E: 0x01,0x02,0x04,0x07,0x08,0x0B,0x0D, and 0x0E.

; 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_PORTE_DATA_R  EQU 0x400243FC
GPIO_PORTE_DIR_R   EQU 0x40024400
GPIO_PORTE_DEN_R   EQU 0x4002451C
SYSCTL_RCGCGPIO_R  EQU 0x400FE608

       THUMB
       AREA    DATA, ALIGN=2
;global variables go here
      ALIGN
      AREA    |.text|, CODE, READONLY, ALIGN=2
      EXPORT  Start
Start
    ;code to run once that initializes PE3,PE2,PE1,PE0
    LDR R0, =SYSCTL_RCGCGPIO_R
    LDR R1, [R0]
    ORR R1, #0x10
    STR R1, [R0]
    NOP
    NOP
    LDR R0, =GPIO_PORTE_DIR_R
    LDR R1, [R0]
    ORR R1, #0x08
    STR R1, [R0]
    LDR R0, =GPIO_PORTE_DEN_R
    LDR R1, [R0]
    ORR R1, #0x0F
    STR R1, [R0]
loop
    ;code that runs over and over
    LDR R0, =GPIO_PORTE_DATA_R
    LDR R1, [R0]
    AND R2, R1, #0x01
    AND R3, R1, #0x02
    LSR R3, #1
    AND R4, R1, #0x04
    LSR R4, #2
    EOR R5, R2, R3
```

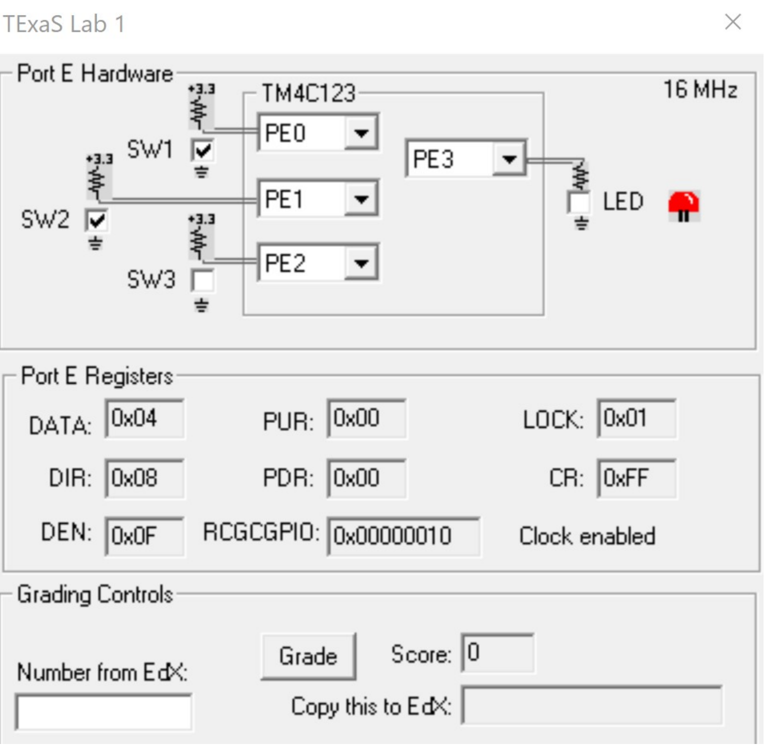
```
        EOR R5, R5, R4
        EOR R5, #1
        LSL R5, #3
        STR R5, [R0]

        B    loop



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

```



Port E Registers

DATA: 0x08
PUR: 0x00
LOCK: 0x01
DIR: 0x08
PDR: 0x00
CR: 0xFF
DEN: 0x0F
RCGCGPIO: 0x00000010
Clock enabled

Number from EdX:
Score: 0
Copy this to EdX:



Port E Registers

DATA: 0x04 PUR: 0x00 LOCK: 0x01

DIR: 0x08 PDR: 0x00 CR: 0xFF

DEN: 0x0F RCGCGPIO: 0x00000010 Clock enabled

Grading Controls

Number from EdX:

Grade Score: 0

Copy this to EdX: