Laboratory Report #5

Name: German E Felisarta III **Group Number:** 3
Laboratory Exercise Title: Parallel I/O Devices Interfacing **Date Completed:** 11/11/2020

ACTIVITY #1

Fig 1a. Circuit Schematic

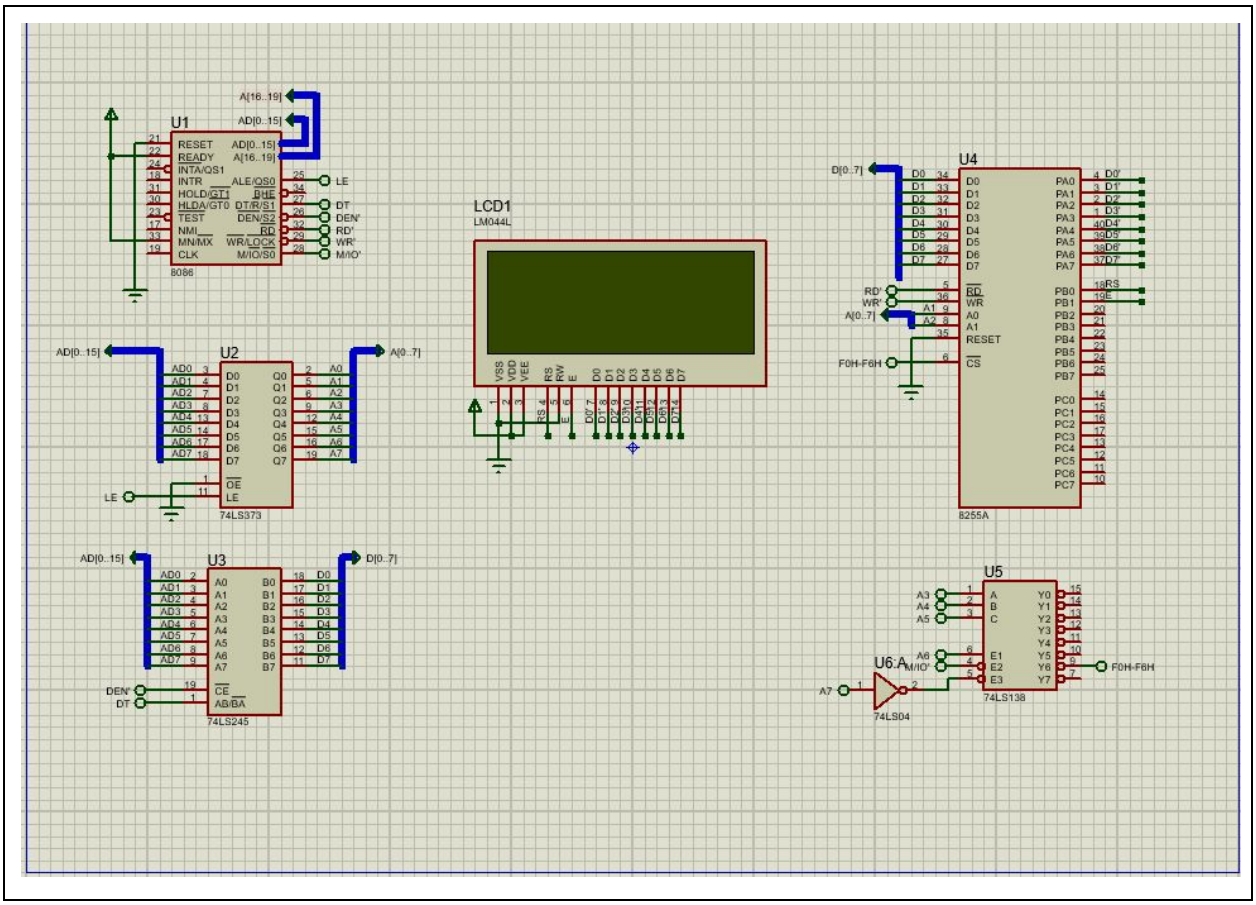


Fig 1b. Display Output

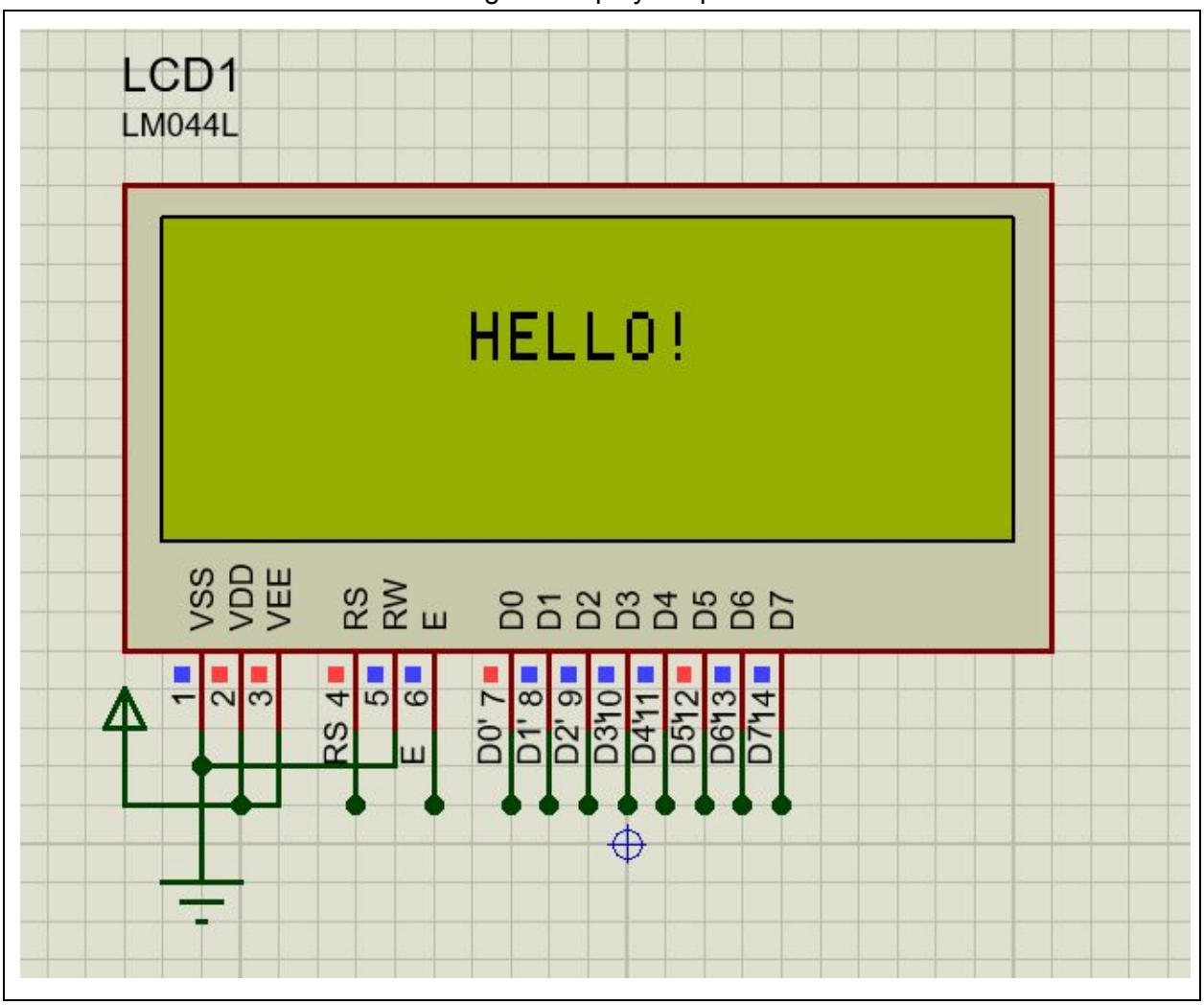


Fig 1c. Source Code

```
;===================================================================
; Main.asm file generated by New Project wizard
;
; Created:   Mon Nov 16 2020
; Processor: 8086
; Compiler:  MASM32
;
; Before starting simulation set Internal Memory Size
; in the 8086 model properties to 0x10000
;
; German E Felisarta III        16101002       CpE3104 Grp 1
;
;===================================================================

DATA SEGMENT

PORTA       EQU 0F0H     ; PORT ADDRESSES
PORTB       EQU 0F2H
PORTC       EQU 0F4H
COM_REG EQU 0F6H


DATA ENDS

CODE    SEGMENT PUBLIC 'CODE'
       ASSUME CS:CODE

START:

  MOV DX, COM_REG         ; STORE COMMAND REGISTER ADDRESS
  MOV AL, 89H
  OUT DX, AL

PRINT:                          ;PRINT HELLO STRING
  CALL INIT_LCD
  MOV AL, 0C6H
  CALL INST_CTRL

  MOV AL, 'H'
  CALL DATA_CTRL
  MOV AL, 'E'
  CALL DATA_CTRL
  MOV AL, 'L'
  CALL DATA_CTRL
  MOV AL, 'L'
```

```
  CALL DATA_CTRL
  MOV AL, 'O'
  CALL DATA_CTRL
  MOV AL, '!'
  CALL DATA_CTRL
  HLT

INIT_LCD PROC NEAR
  MOV AL, 38H
  CALL INST_CTRL
  MOV AL, 0CH
  CALL INST_CTRL
  MOV AL, 01H
  CALL INST_CTRL
  MOV AL, 06H
  CALL INST_CTRL
 INIT_LCD ENDP

INST_CTRL PROC NEAR
  MOV DX, PORTA
  OUT DX, AL
  MOV DX, PORTB
  MOV AL, 02H
  OUT DX, AL
  CALL DELAY

  MOV DX, PORTB
  MOV AL, 0H
  OUT DX, AL
INST_CTRL ENDP

DATA_CTRL PROC NEAR
  MOV DX, PORTA
  OUT DX, AL
  MOV DX, PORTB
  MOV AL, 03H
  OUT DX, AL
  CALL DELAY
  MOV DX, PORTB
  MOV AL, 01H
  OUT DX, AL


DATA_CTRL ENDP

DELAY PROC NEAR
```

```
  MOV CX, 0FFFH

  DELAY_LOOP:

    DEC CX
    CMP CX, 00H
    JNZ DELAY_LOOP

  RET


DELAY ENDP

CODE ENDS
END
```

ACTIVITY #2

Fig 2a. Circuit Schematic

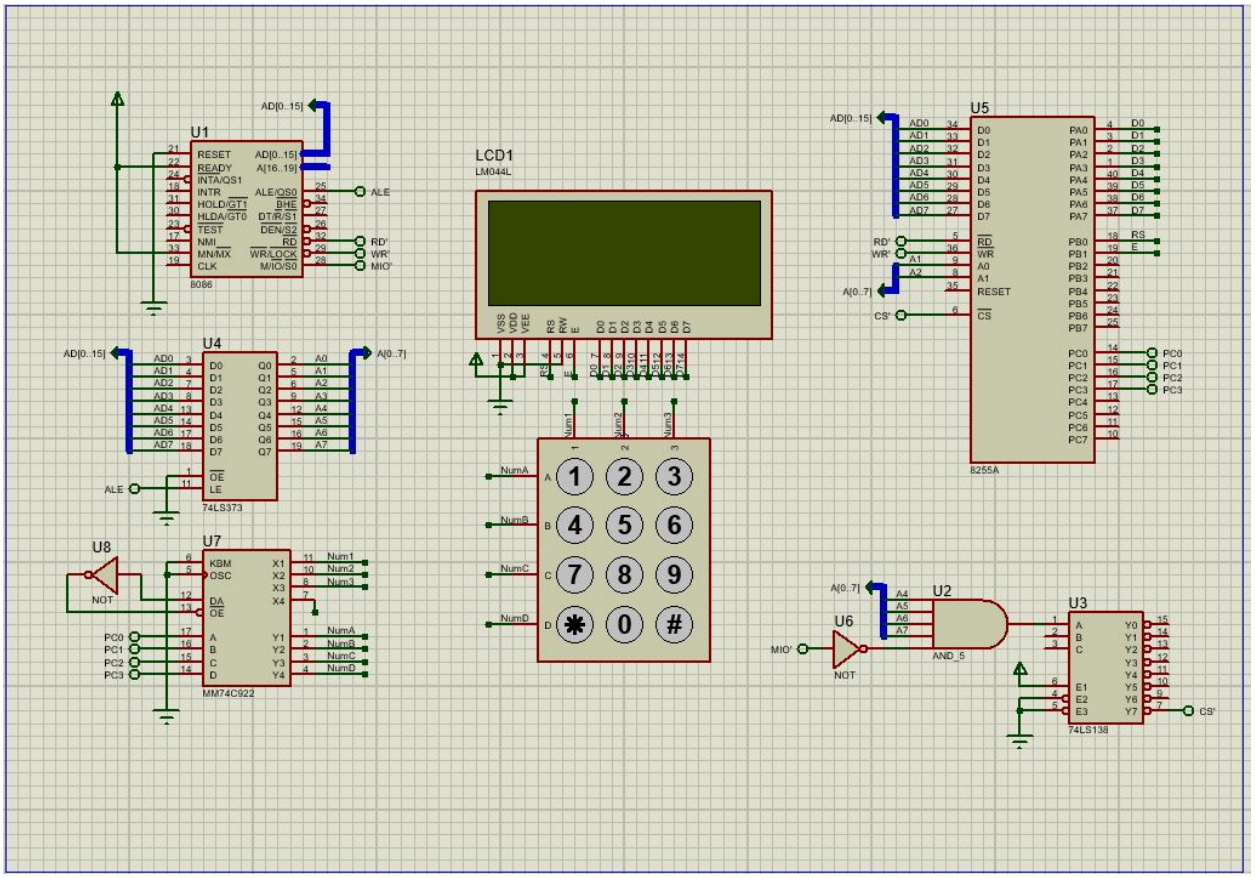
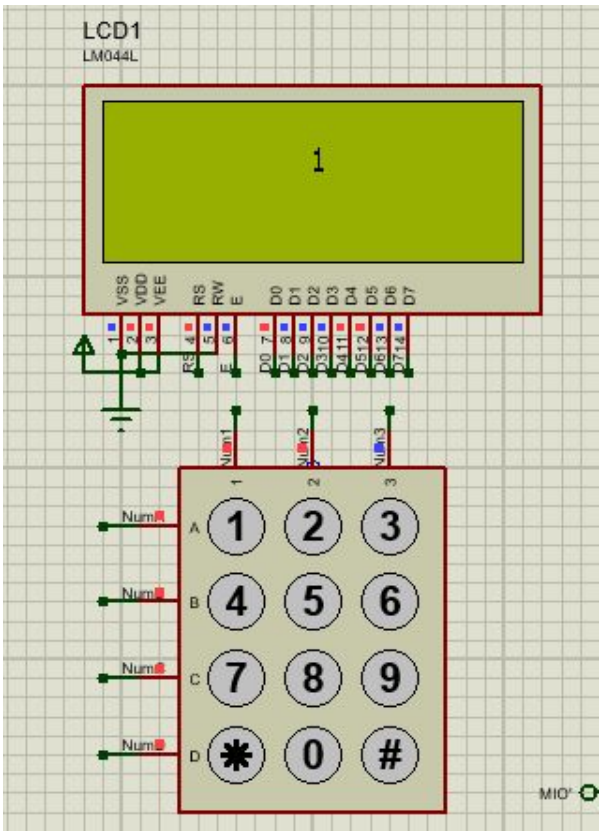
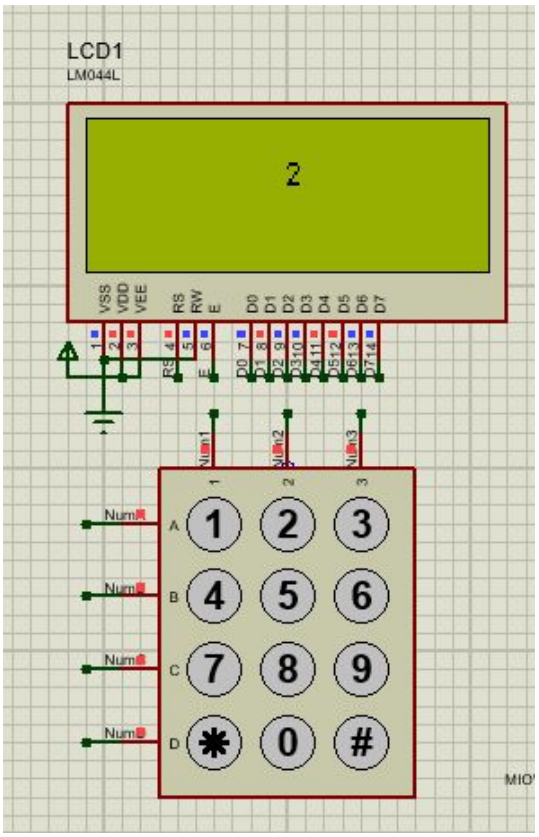
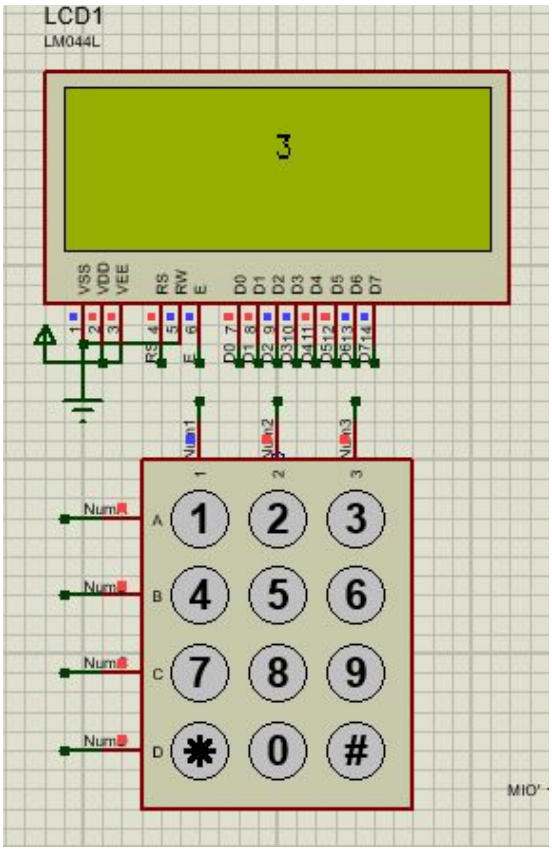
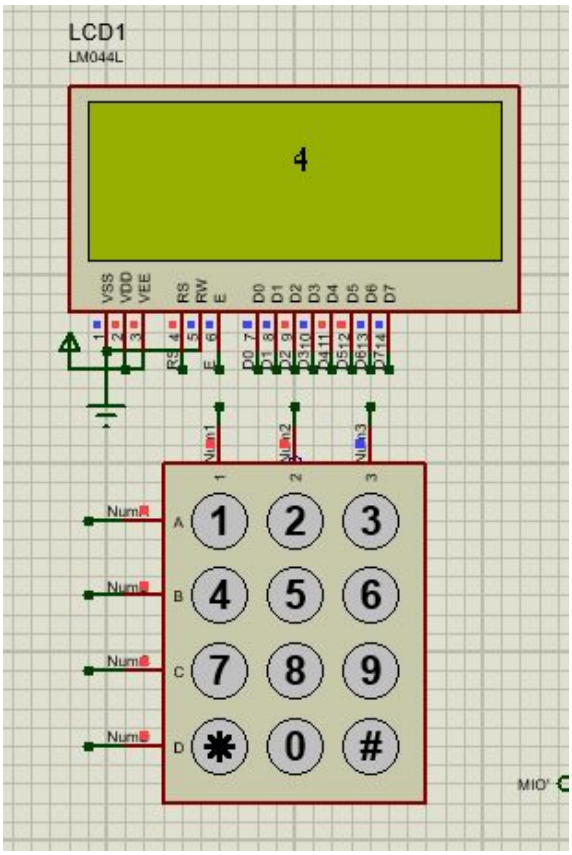


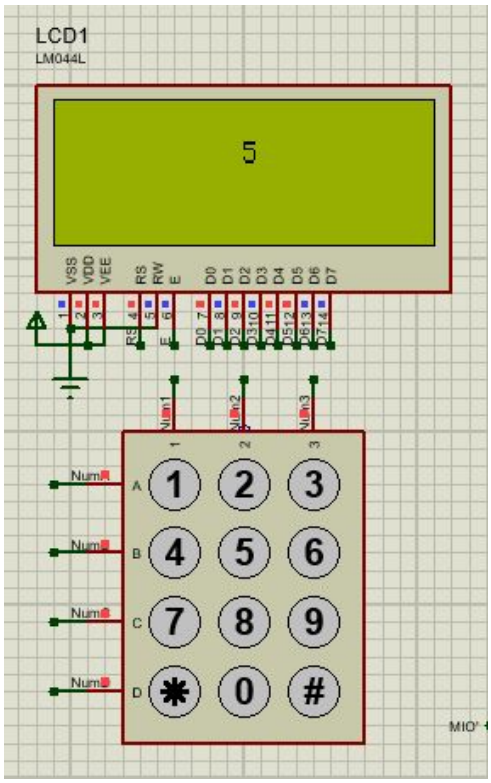
Fig 2b. Display Outputs

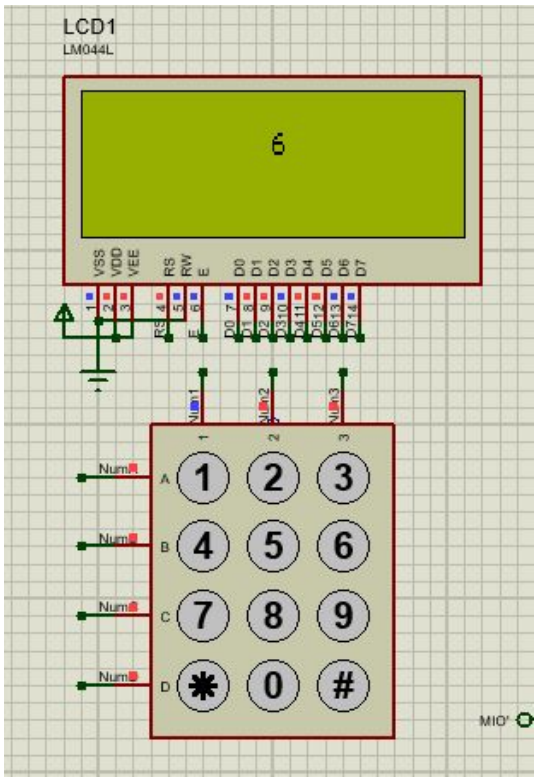

LCD1
LM044L
1
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
Num A 1 2 3
Num B 4 5 6
Num C 7 8 9
Num D * 0 #

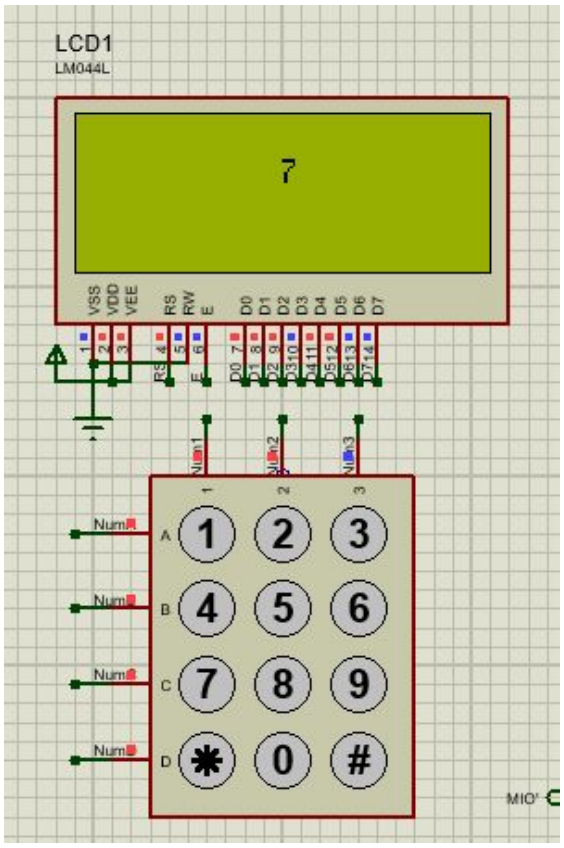

LCD1
LM044L
2
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
Num A 1 2 3
Num B 4 5 6
Num C 7 8 9
Num D * 0 #

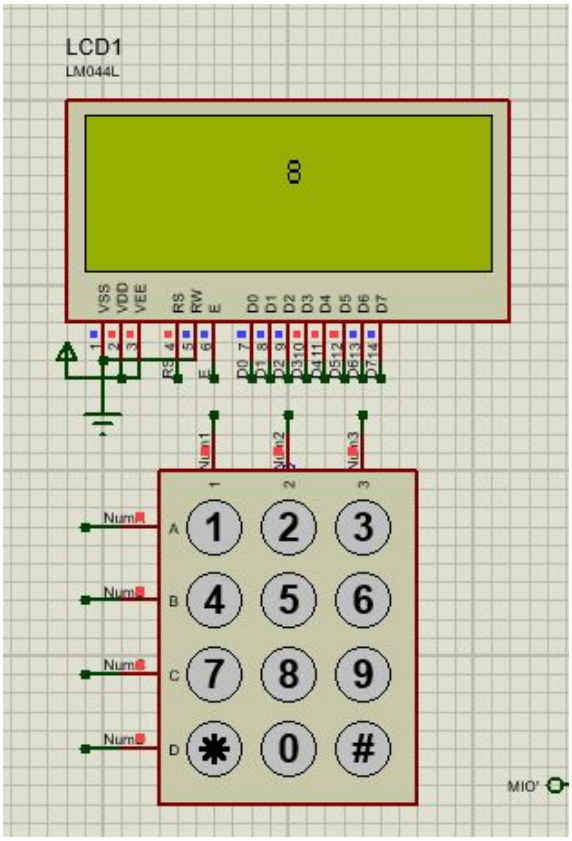

LCD1
LM044L
3
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
NumA NumB NumC NumD
A 1 2 3
B 4 5 6
C 7 8 9
D * 0 #

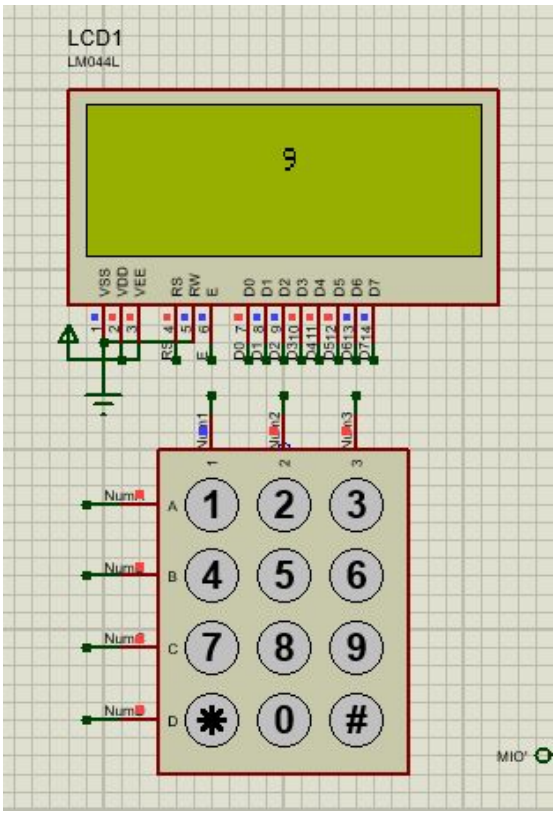

LCD1
LM044L
4
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
NumA NumB NumC NumD
A 1 2 3
B 4 5 6
C 7 8 9
D * 0 #


LCD1
LM044L
5
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
A 1 2 3
B 4 5 6
C 7 8 9
D * 0 #


LCD1
LM044L
6
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
A 1 2 3
B 4 5 6
C 7 8 9
D * 0 #


LCD1
LM044L
7
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
A 1 2 3
B 4 5 6
C 7 8 9
D * 0 #
MIO


LCD1
LM044L
8
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
A 1 2 3
B 4 5 6
C 7 8 9
D * 0 #
MIO


LCD1
LM044L
9
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
A 1 2 3
B 4 5 6
C 7 8 9
D * 0 #
MIO

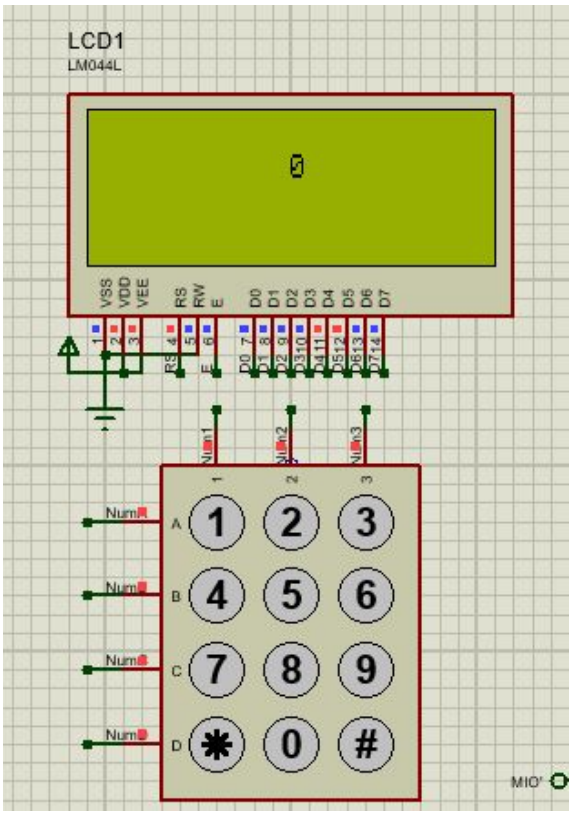

LCD1
LM044L
0
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
A 1 2 3
B 4 5 6
C 7 8 9
D * 0 #
MIO

Fig 2c. Source Code

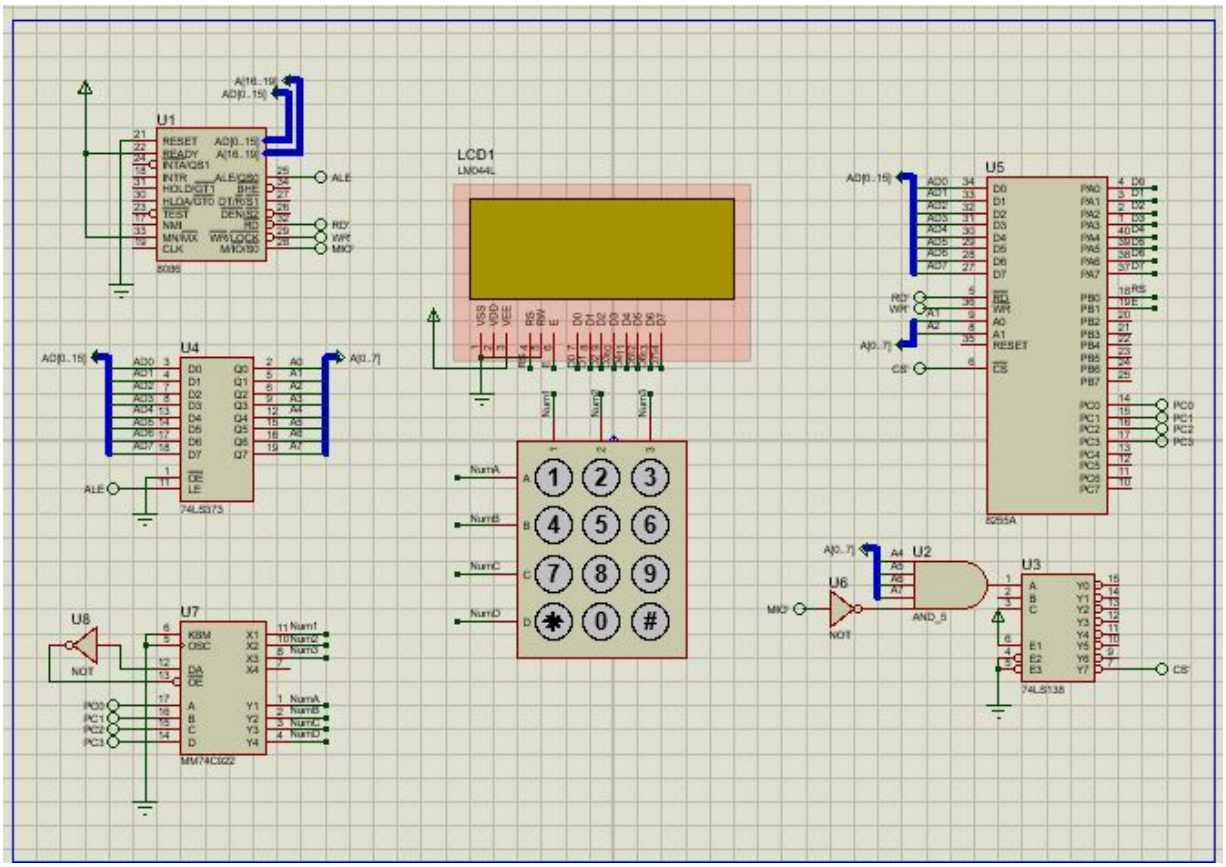
```
;===================================================================
; Main.asm file generated by New Project wizard
;
; Created:   Mon Nov 16 2020
; Processor: 8086
; Compiler:  MASM32
;
; Before starting simulation set Internal Memory Size
; in the 8086 model properties to 0x10000
;
; German E Felisarta III          16101002        CpE3104 Grp 1
;
;===================================================================
DATA SEGMENT
  PORTA EQU 0F0H ; PORT ADDRESSES
  PORTB EQU 0F2H
  PORTC EQU 0F4H
  COM_REG EQU 0F6H
DATA ENDS

CODE    SEGMENT PUBLIC 'CODE'
      ASSUME CS:CODE

START:
  MOV DX, COM_REG           ; STORE COMMAND REGISTER ADDRESS
  MOV AL, 89H
  OUT DX, AL
  CALL INIT_LCD

MAIN:
  CALL KEYPRESS
  MOV AH, AL
  MOV AL, 01H        ; CLEAR
  CALL INST_CTRL
  MOV AL, 0CAH
  CALL INST_CTRL
  MOV AL, AH
  XOR AL, 0C0H
  CALL DATA_CTRL     ; DISPLAY TO LCD
  JMP MAIN           ; LOOP

INST_CTRL PROC NEAR
   MOV DX, PORTA
   OUT DX, AL
   MOV DX, PORTB
```

```
    MOV AL, 02H
    OUT DX, AL
    CALL DELAY
    MOV DX, PORTB
    MOV AL, 00H
    OUT DX, AL
    RET
INST_CTRL ENDP

INIT_LCD PROC NEAR
    MOV AL, 38H
    CALL INST_CTRL
    MOV AL, 0CH
    CALL INST_CTRL
    MOV AL, 01H
    CALL INST_CTRL
    MOV AL, 06H
    CALL INST_CTRL
    RET
INIT_LCD ENDP

DATA_CTRL PROC NEAR
    MOV DX, PORTA
    OUT DX, AL
    MOV DX, PORTB
    MOV AL, 03H
    OUT DX, AL
    CALL DELAY
    MOV DX, PORTB
    MOV AL, 01H
    OUT DX, AL
    RET
DATA_CTRL ENDP

DELAY PROC NEAR          ; TIME DELAY (optional)
  MOV CX, 0FFFh
  DELAY_LOOP:
     DEC CX
     CMP CX, 00H
     JNZ DELAY_LOOP
  RET
DELAY ENDP

KEYPRESS PROC
  CHECK:              ; INPUT FOR NUMPAD
     XOR AL, AL
```

```
    IN AL, PORTC
    CMP AL, 0FFH
    JE CHECK
    CALL ADJUST     ; ADJUST ADJUSTS NUMPAD BINDINGS TO MATCH MM74C
  RET
KEYPRESS ENDP

ADJUST PROC
  CMP AL, 0F0H
  JE INCREMENT
  CMP AL, 0F1H
  JE INCREMENT
  CMP AL, 0F2H
  JE INCREMENT
  CMP AL, 0F8H
  JE DECREMENT
  CMP AL, 0F9H
  JE DECREMENT
  CMP AL, 0FAH
  JE DECREMENT
  CMP AL, 0FDH
  JE ZERO
  JMP LOOP_END

  INCREMENT:
    INC AL
    JMP LOOP_END
  DECREMENT:
    DEC AL
    JMP LOOP_END
  ZERO:
    MOV AL, 0F0H
  LOOP_END:
    RET
ADJUST ENDP

CODE ENDS
     END START
```

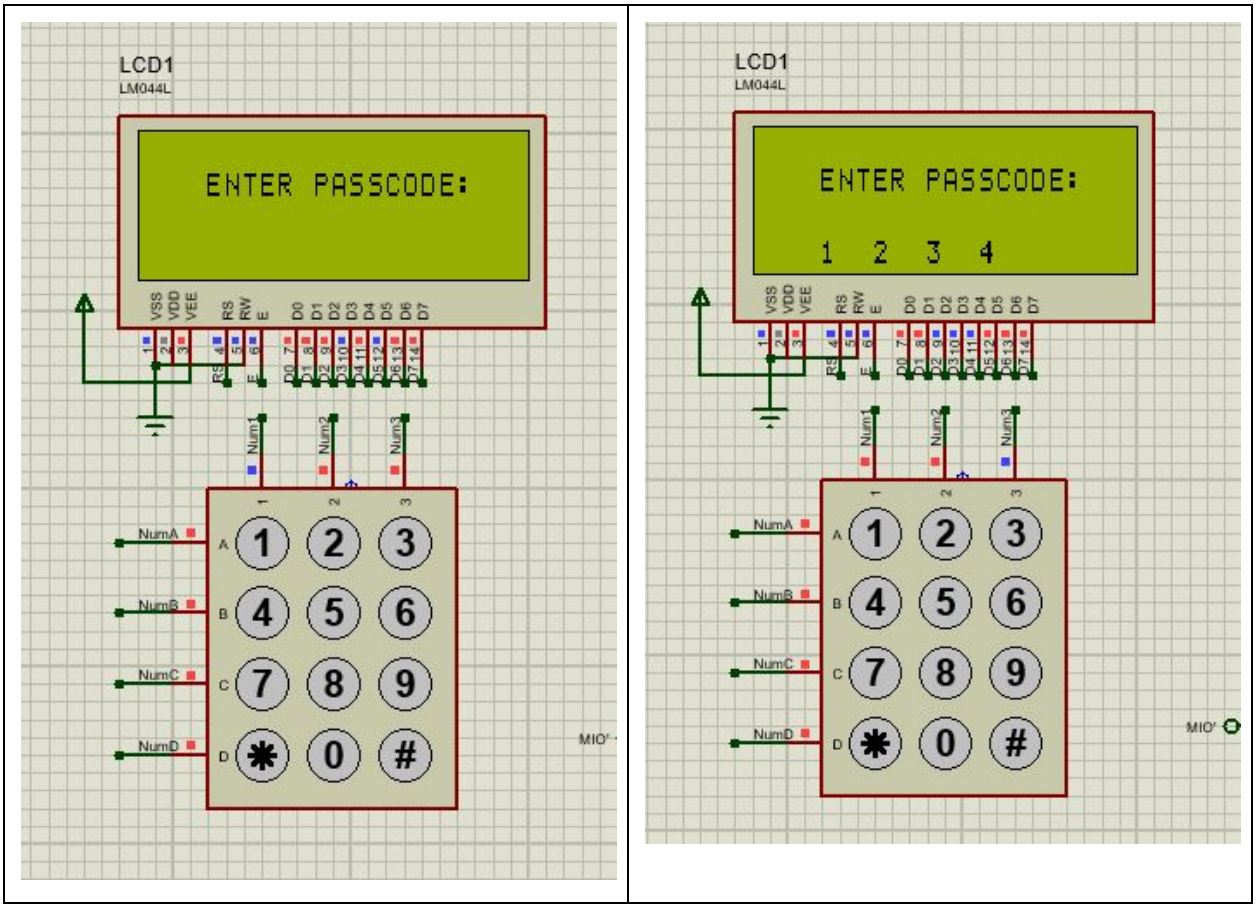
ACTIVITY #3

Fig 3a. Circuit Schematic



*Proteus 8.6 bug, LCD not showing properly. It worked the first time. Also shows up only when run

Fig 3b. Display Outputs


LCD1
LM044L
ENTER PASSCODE:
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
NumA NumB NumC NumD
1 2 3
4 5 6
7 8 9
* 0 #
LCD1
LM044L
ENTER PASSCODE:
1 2 3 4
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
Num1 Num2 Num3
NumA NumB NumC NumD
1 2 3
4 5 6
7 8 9
* 0 #

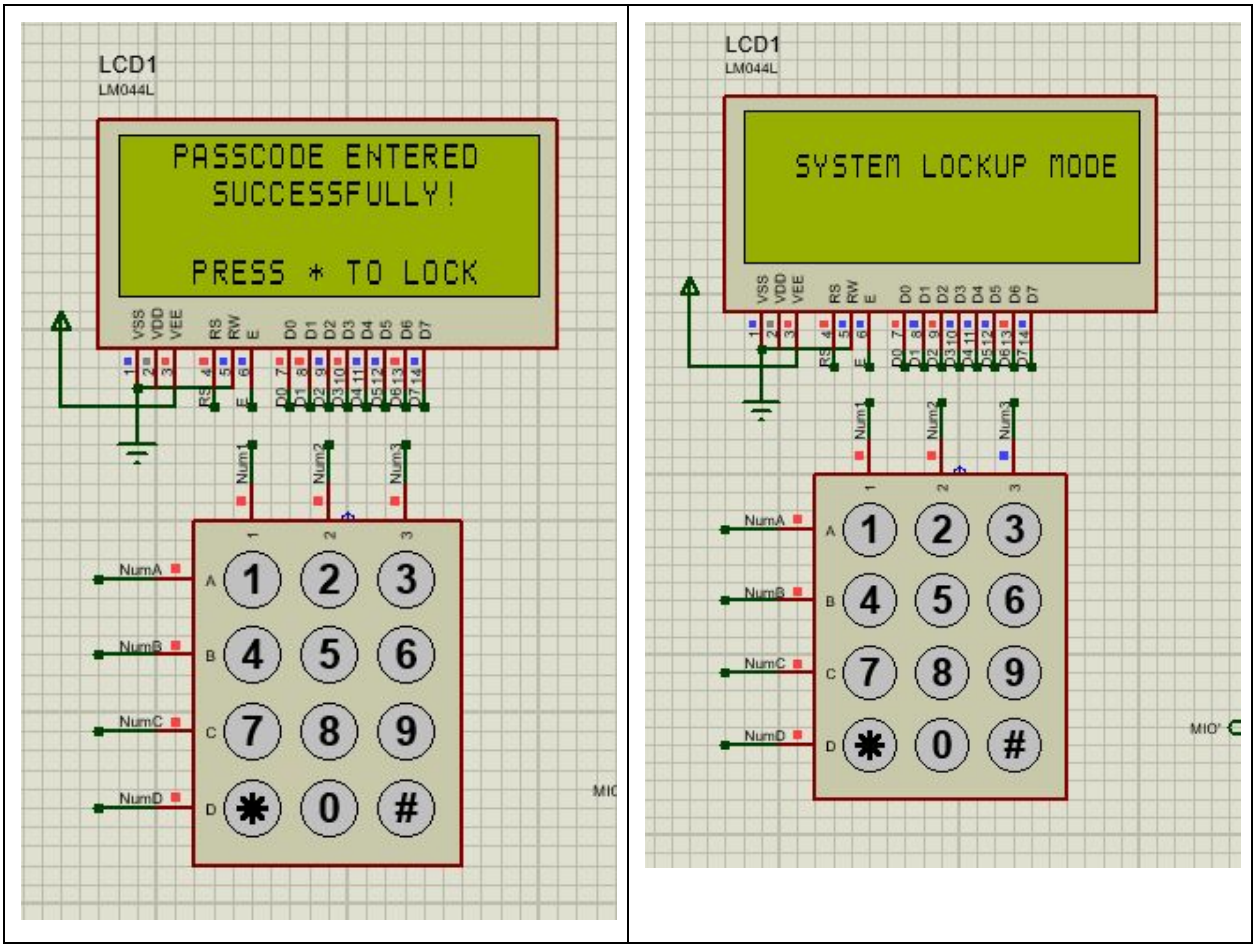


Fig 3c. Source Code

```
;===================================================================
; Main.asm file generated by New Project wizard
;
; Created:   Mon Nov 16 2020
; Processor: 8086
; Compiler:  MASM32
;
; Before starting simulation set Internal Memory Size
; in the 8086 model properties to 0x10000
;
; German E Felisarta III 16101002   CpE3104 Grp1
;
;===================================================================

DATA SEGMENT
  PORTA EQU 0F0H                                  ; PORT ADDRESSES
```

```
  PORTB EQU 0F2H
  PORTC EQU 0F4H
  COM_REG EQU 0F6H
  PASSWORD db '13579', '$'                    ; PASSWORD
  PROMPT_REQ db 'ENTER PASSCODE:', '$'        ; DISPLAY PROMPTS
  MSG1 db 'PASSCODE ENTERED', '$'
  MSG2 db 'SUCCESSFULLY!', '$'
  MSG3 db 'PRESS * TO LOCK', '$'
  PROMPT_LOCKDOWN db 'SYSTEM LOCKUP MODE', '$'
DATA ENDS

CODE    SEGMENT PUBLIC 'CODE'
       ASSUME CS:CODE

START:
  MOV DX, COM_REG
  MOV AL, 89H
  OUT DX, AL
  CALL INIT_LCD
  MOV BH, 03H
  LEA DI, PASSWORD

MAIN:
  MOV AL, 01H                    ; CLEAR
  CALL INST_CTRL
  MOV AL, 0C3H
  CALL INST_CTRL
  LEA SI, PROMPT_REQ
  CALL DISP_STRING
  CALL INPUT_PASS
  HLT

INST_CTRL PROC NEAR
   MOV DX, PORTA
   OUT DX, AL
   MOV DX, PORTB
   MOV AL, 02H
   OUT DX, AL
   CALL DELAY
   MOV DX, PORTB
   MOV AL, 00H
   OUT DX, AL
   RET
INST_CTRL ENDP

INIT_LCD PROC NEAR
```

```
   MOV AL, 38H
   CALL INST_CTRL
   MOV AL, 0CH          ; ACTIVATE DISPLAY
   CALL INST_CTRL
   MOV AL, 01H          ; CLEAR
   CALL INST_CTRL
   MOV AL, 06H                    ; MOVE CURSOR
   CALL INST_CTRL
   RET
INIT_LCD ENDP

DATA_CTRL PROC NEAR
   MOV DX, PORTA
   OUT DX, AL
   MOV DX, PORTB
   MOV AL, 03H
   OUT DX, AL
   CALL DELAY
   MOV DX, PORTB
   MOV AL, 01H
   OUT DX, AL
   RET
DATA_CTRL ENDP

DELAY PROC NEAR                ; TIME DELAY (optional)
  MOV CX, 0FFFh
  DELAY_LOOP:
    DEC CX
    CMP CX, 00H
    JNZ DELAY_LOOP
  RET
DELAY ENDP

KEYPRESS PROC
  CHECK:                       ; INPUT FOR NUMPAD
    XOR AL, AL
    IN AL, PORTC
    CMP AL, 0FFH
    JE CHECK
    CALL ADJUST              ; ADJUST ADJUSTS NUMPAD BINDINGS TO MATCH MM74C
  RET
KEYPRESS ENDP

ADJUST PROC
  CMP AL, 0F0H
  JE INCREMENT
```

```
  CMP AL, 0F1H
  JE INCREMENT
  CMP AL, 0F2H
  JE INCREMENT
  CMP AL, 0F8H
  JE DECREMENT
  CMP AL, 0F9H
  JE DECREMENT
  CMP AL, 0FAH
  JE DECREMENT
  CMP AL, 0FDH
  JE ZERO
  JMP LOOP_END

  INCREMENT:
    INC AL
    JMP LOOP_END
  DECREMENT:
    DEC AL
    JMP LOOP_END
  ZERO:
    MOV AL, 0F0H
  LOOP_END:
    RET
ADJUST ENDP

DISP_STRING PROC NEAR              ; FOR PRINT STRING
  DISP_LOOP:                    ; LOOP TO PRINT EACH CHARACTER FROM STRING
    MOV AL, [SI]
    CMP AL, '$'
    JE EXIT
    CALL DATA_CTRL
    CALL delay
    INC SI
    JMP DISP_LOOP
  EXIT:
    RET
DISP_STRING ENDP

INPUT_PASS PROC
  MOV BL, 0D7H
  AUTH_LOOP:
    MOV AL, BL                    ; SET CURSOR ADDRESS
    CALL INST_CTRL
    CALL KEYPRESS
    XOR AL, 0C0H
```

```
    CALL CHECK
    CALL DATA_CTRL
    ADD BL, 03H             ; ADJUST CURSOR ADDRESS
    CALL AWAIT_INPUT
    JMP AUTH_LOOP           ; LOOP FOR NEXT CHARACTER INPUT
  RET
INPUT_PASS ENDP

AWAIT_INPUT PROC                ; LOOP UNTIL INPUT IS COMPLETE
  CHECK_STATUS:
    IN AL, PORTC
    CMP AL, 0FFH
    JNE CHECK_STATUS
  RET
AWAIT_INPUT ENDP

CHECK PROC                      ; CHECK ENTERED PASSWORD
  CMP AL, [DI]
  JNE INVALID               ; IF INPUT IS INVALID, RESTART SYSTEM
  PUSH BX
  MOV BL, '$'
  INC DI
  CMP [DI], BL              ; CHECKS IF ALL CHARACTERS INPUTTED IS CORRECT
  POP BX
  JE VALID
  RET

  INVALID:
    dec bh
    cmp bh, 00h
    je LOCKDOWN
    lea di, PASSWORD        ; RESET INDEX
    jmp MAIN                ; RESTARTS SYSTEM BECAUSE INVALID INPUT

  VALID:
    MOV AL, 01H             ; CLEAR
    CALL INST_CTRL
    MOV AL, 082H
    CALL INST_CTRL
    LEA si, MSG1            ; PROMPT
    CALL DISP_STRING
    MOV AL, 0C4H
    CALL INST_CTRL
    LEA si, MSG2            ; PROMPT
    CALL DISP_STRING
    MOV AL, 0D7H
```

```
    CALL INST_CTRL
    LEA si, MSG3                ; PROMPT
    CALL DISP_STRING


   CHECK_STAR:                  ; LOOP WHEN '*' IS NOT INPUTTED
    CALL KEYPRESS
    CMP AL, 0FCH                ; CHECK FOR STAR INPUT
    JNE CHECK_STAR
    MOV BH, 03h                 ; DEFAULT VALUES
    LEA DI, PASSWORD            ; RESTART SYSTEM
    JMP MAIN

  LOCKDOWN:
    MOV AL, 01H                 ; CLEAR
    CALL INST_CTRL
    MOV AL, 0CH
    CALL INST_CTRL
    MOV AL, 0C2H
    CALL INST_CTRL
    LEA si, PROMPT_LOCKDOWN
    CALL DISP_STRING
    HLT
CHECK ENDP

CODE ENDS
     END START
```