;DEFINING PINS AND THEIR CORRESPONDING FUNCTIONS FIRST

RS EQU P2.0 ;RS PIN OF LCD

RW EQU P2.1 ;RW PIN OF LCD

E EQU P2.2 ;ENBL PIN OF LCD

MOTOR EQU P2.3 ;MOTOR DRIVER PIN 1

REV\_MOTOR EQU P2.4 ;MOTOR DRIVER PIN 2

PASSWORD\_NO\_DIGITS EQU 4 ;NUMBER OF DIGITS IN THE PASSWORD

PIN\_RESET\_KEY EQU '4'

ON\_KEY EQU '5'

OFF\_KEY EQU '6'

REV\_KEY EQU '4'

;ROWS OF KEYPAD:

ROW\_A EQU P3.4

ROW\_B EQU P3.5

ROW\_C EQU P3.6

ROW\_D EQU P3.7

;COLUMNS OF KEYPAD:

COL\_1 EQU P3.0

COL\_2 EQU P3.1

COL\_3 EQU P3.2

COL\_4 EQU P3.3

ORG 00H ;CODE STARTS FROM HERE ON

MOV SP, #70H

MOV PSW, #00H

;INITIALIZING INPUT & OUTPUT PORTS

MOV P0,#00H

MOV P3,#0FH

MOV P2,#00H

MOV R1, #50H ;POINTER FOR STORING CORRECT PASSWORD

MOV R3,#PASSWORD\_NO\_DIGITS ;LOOP COUNTER FOR COPYING PASSWORD

MOV DPTR,#REAL\_PASSWORD ;COPYING STORED REAL PASSWORD ADDRESS LOCATION

COPY\_PASSWORD: ;COPYING THE REAL PASSWORD TO RAM LOCATIONS STARTING FROM 50H

CLR A

MOVC A,@A+DPTR

MOV @R1,A

INC R1

INC DPTR

DJNZ R3,COPY\_PASSWORD

START: MOV DPTR, #MYLCD ;INITIALIZE AND START THE LCD

LCD\_IN: CLR A

MOVC A,@A+DPTR

ACALL COMNWRT

ACALL DELAY

JZ LOAD\_WELCOME

INC DPTR

SJMP LCD\_IN

LOAD\_WELCOME: ;LOAD THE WELCOME MESSAGE

MOV DPTR,#WELCOME

ACALL TXT\_LED

LCALL DELAY\_H

MOV R7,#0 ;FOR COUNTING THE NUMBER OF TIMES INCORRECT PASSWORD IS ENTERED

INITIALIZATION:

CLR A

MOV R3,A ;INITIALLY SET R3 AND R4 AS ZERO

MOV R4,A

MOV R5, #PASSWORD\_NO\_DIGITS ;NUMBER OF DIGITS IN THE PASSWORD

MOV R0, #40H ;POINTER 1 FOR CHECKING PASSWORD IS CORRECT OR NOT

MOV R1, #50H ;POINTER 2 FOR CHECKING PASSWORD IS CORRECT OR NOT

CONTINUE:

LCALL CLEAR\_DISPLAY\_FINAL ;CLEAR LCD DISPLAY

MOV DPTR,#PASSWORD ;PROMPT USER FOR ENTERING THE PASSWORD

ACALL TXT\_LED

GO\_: MOV A,#8BH ;MOVE CURSOR TO CORRECT POSITION

ACALL COMNWRT

ACALL DELAY

KEYPAD: ACALL KEYBOARD ;KEYBOARD SUBROUTINE ACTIVATED AND CALLED

PASSWORD\_STORE:

CLR A

MOVC A,@A+DPTR

MOV @R0,A ;STORE ENTERED PASSWORD IN RAM LOCATIONS

MOV A,#'\*' ;SHOW ENTERED PASSWORD IN WITH '\*' SIGN

ACALL DATAWRT ;CALL DISPLAY SUBROUTINE

ACALL DELAY

INC R0 ;INCREMENT THE POINTER

DJNZ R5,NEXT ;UNLESS R5 BECOMES ZERO, TAKE THE NEXT PASSWORD DIGIT INPUT

SJMP DO ;IF ENTIRE PASSWORD ENTERED THEN JUMP TO DO

NEXT: SJMP KEYPAD

DO: MOV R5,#0 ;COUNT THE NUMBER OF ERROR IN DIGITS

MOV R2,#PASSWORD\_NO\_DIGITS ;LOOP COUNTER FOR PASSWORD\_CHECK

MOV R0,#40H

MOV R1,#50H

REAL\_PASSWORD\_CHECK: ;CHECKING IF THE ENTERED PASSWORD IS CORRECT OR NOT

MOV A,@R0

MOV B,A

CLR A

MOV A,@R1

CJNE A,B,COUNT

SJMP NOT\_COUNT

COUNT: INC R5 ;IF THE ENTERED PASSWORD DOESN'T MATCH COUNT THE NUMBER OF DIGITS IN ERROR

NOT\_COUNT:

INC R0

INC R1

DJNZ R2,REAL\_PASSWORD\_CHECK

CJNE R5,#0,NEXT2 ;IF R5 = 0, NO ERROR IN ENTERED PASSWORD

SJMP GRANT ; THEN GRANT PERMISSION TO USER FOR CONTROLLING THE MOTOR

NEXT2: CJNE R5,#1,DENY ;IF NUMBER OF DIGITS IN ERROR IS 1, WE ARE CLOSE

;OTHERWISE WE SHOW ACCESS DENIED

VERY\_CLOSE:

ACALL NEW\_LINE

MOV DPTR,#CLOSE ;SHOWING THAT THEY ARE CLOSE TO THE REAL PASSWORD

ACALL TXT\_LED

SJMP CHECK\_ATTEMPTS ;CHECKING IN THE 3RD ATTEMPT IS REACHED OR NOT

GRANT: ACALL NEW\_LINE

MOV DPTR,#GRANTED ;WE SHOW THAT THE USER HAS BEEN GRANTED THE PERMISSION WHEN PASSWORD IS CORRECT

ACALL TXT\_LED

SJMP SUCCESS

DENY: ACALL NEW\_LINE

MOV DPTR,#DENIED ;WE SHOW THAT THE ACCESS IS DENIED WHEN THE ENTERED PASSWORD IS NOT CORRECT AND NOT CLOSE AS WELL

ACALL TXT\_LED

SJMP CHECK\_ATTEMPTS ;CHECKING IN THE 3RD ATTEMPT IS REACHED OR NOT

CHECK\_ATTEMPTS: ;CHECKING IF MAX NUMBER OF ATTEMPTS IS REACHED OR NOT

INC R7 ;INCREMENTING R7, FOR EACH ATTEMPT

CJNE R7,#3,NEXTRY ;IF R7 IS EQUAL TO 3, JUMP TO NEXTRY

SJMP LOCK ;IF R7=3, LOCK THE SYSTEM

NEXTRY:

LJMP INITIALIZATION ;START FROM BEGINNING, ASKING USER TO ENTER PASSWORD IN NEXT ATTEMPT

LOCK: LCALL CLEAR\_DISPLAY\_FINAL ;DISPLAYING THAT THE SYSTEM IS LOCKED

MOV DPTR,#LOCKED

ACALL TXT\_LED

ACALL NEW\_LINE

MOV DPTR,#RESET\_PIN ;SHOWING THE INSTRUCTION TO RESET THE PIN

ACALL TXT\_LED

ACALL KEYBOARD

;NEW\_PASSWORD

CHECK\_IF\_4\_PRESSED: ;CHECKING IF 4 IS PRESSED, IF NOT THEN SYSTEM REMAINS LOCKED

CLR A

MOVC A,@A+DPTR

MOV R4,A

CJNE R4,#PIN\_RESET\_KEY,LOCK ;IF RESET KEY IS NOT PRESSED THEN THE SYTEM WILL REMAIN IN LOCKED STATE

YES\_4\_PRESSED:

MOV R1,#50H ;USE R1 POINTER TO STORE THE NEW PASSWORD

MOV R5,#PASSWORD\_NO\_DIGITS ;LOOP COUNTER FOR NEW PASSWORD ENTER

PROMPT\_NEW\_PASSWORD:

LCALL CLEAR\_DISPLAY\_FINAL

MOV DPTR,#NEW\_PASSWORD ;PROMPT NEW PASSWORD FOR THE USER TO ENTER

ACALL TXT\_LED

KEYPAD1:ACALL KEYBOARD ;TAKE INPUT

STORE\_NEW\_PASSWORD:

CLR A

MOVC A,@A+DPTR

MOV @R1,A

MOV A,#'\*'

ACALL DATAWRT ;CALL DISPLAY SUBROUTINE

ACALL DELAY

INC R1

DJNZ R5,NEXT\_ ;UNLESS R5 BECOMES ZERO, TAKE THE NEXT PASSWORD DIGIT INPUT

SJMP DONE\_NEW\_PIN\_INPUT ;IF ENTIRE PASSWORD ENTERED THEN JUMP TO DONE\_NEW\_PIN\_INPUT

NEXT\_: LJMP KEYPAD1 ;OTHERWISE CONTINUE TAKING THE PASSWORD INPUT

DONE\_NEW\_PIN\_INPUT:

LJMP START ;ONCE NEW PASSWORD IS ENTERED, WE GO TO THE BEGINNING, ASKING THE USER TO ENTER THE PASSWORD AND ENTER THE SYSTEM

SUCCESS: ;WE REACH HERE IF THE ENTERED PASSWORD BY USER IS CORRECT

LCALL CLEAR\_DISPLAY\_FINAL

MOV DPTR,#INSTRUCT\_CONTROL ;SHOWING THE CONTROL INSTRUCTIONS FOR THE MOTOR

ACALL TXT\_LED

ACALL NEW\_LINE

MOV DPTR,#MOTOR\_CONTROL ;SHOWING THE NEXT SET OF CONTROL INSTRUCTIONS FOR THE MOTOR

ACALL TXT\_LED

KEYPAD2:ACALL KEYBOARD ;TAKING INPUT FROM THE USER THTORUHG KEYPAD ALLOWING THEM TO CONTROL THE MOTOR

CHECK\_IF\_ON\_PRESSED: ;CHECKING IF THE ON BUTTON IS PRESSED OR NOT I.E. '5'

CLR A

MOVC A,@A+DPTR

MOV R4,A

CJNE R4,#ON\_KEY,CHECK\_IF\_OFF\_PRESSED ;IF ON KEY NOT PRESSED, THEN GO TO CHECK IF OFF PRESSED

YES\_ON\_PRESSED: ;COMES HERE ONLY IF '5' IS PRESSED

CLR REV\_MOTOR

SETB MOTOR

LJMP KEYPAD2 ;ALLOWING THE USER TO ENTER THE NEXT CONTROL INPUT THROUGH KEYPAD

CHECK\_IF\_OFF\_PRESSED: ;CHECKING IF THE OFF BUTTON IS PRESSED OR NOT I.E. '6'

CLR A

MOVC A,@A+DPTR

MOV R4,A

CJNE R4,#OFF\_KEY,CHECK\_IF\_REV\_PRESSED

YES\_OFF\_PRESSED: ;COMES HERE ONLY IF '6' IS PRESSED

CLR MOTOR

CLR REV\_MOTOR

LJMP KEYPAD2 ;ALLOWING THE USER TO ENTER THE NEXT CONTROL INPUT THROUGH KEYPAD

CHECK\_IF\_REV\_PRESSED: ;CHECKING IF THE REVERSE BUTTON IS PRESSED OR NOT I.E. '4'

CLR A

MOVC A,@A+DPTR

MOV R4,A

CJNE R4,#REV\_KEY,WRONG\_PRESS

YES\_REV\_PRESSED: ;COMES HERE ONLY IF '4' IS PRESSED

CPL MOTOR

CPL REV\_MOTOR

LJMP KEYPAD2 ;ALLOWING THE USER TO ENTER THE NEXT CONTROL INPUT THROUGH KEYPAD

WRONG\_PRESS: ;COMES HERE ONLY WRONG KEY IS PRESSED, SO NOTHING HAPPENS TO MOTOR

LJMP KEYPAD2 ;ALLOWING THE USER TO ENTER THE CONTROL INPUT THROUGH KEYPAD AGAIN

FINISH: SJMP FINISH

;SUBROUTINES ARE DEFINED FORM HERE ON

ORG 400H

;FIRST THE KEBOARD SUBROUTINE

KEYBOARD:

K1: CLR ROW\_A ;FIRST CHECK IF NO KEY IS PRESSED IN THE KEYBOARD

CLR ROW\_B

CLR ROW\_C

CLR ROW\_D

MOV A,P3

ANL A,#00001111B

CJNE A,#00001111B,K1

K2: ACALL DELAY ;NOW CHECK IF KEY IS PRESSED IN THE KEYBOARD

MOV A,P3

ANL A,#00001111B

CJNE A,#00001111B,K3

SJMP K2

K3: ACALL DELAY ;AGAIN CHECK IF KEY IS PRESSED IN THE KEYBOARD

MOV A,P3

ANL A,#00001111B

CJNE A,#00001111B,K4

SJMP K2

K4: CLR ROW\_A ;CHECKING IF KEY IS PRESSED FROM ROW\_A

SETB ROW\_B

SETB ROW\_C

SETB ROW\_D

MOV A,P3

ANL A,#00001111B

CJNE A,#00001111B,ROW\_0

SETB ROW\_A ;CHECKING IF KEY IS PRESSED FROM ROW\_B

CLR ROW\_B

SETB ROW\_C

SETB ROW\_D

MOV A,P3

ANL A,#00001111B

CJNE A,#00001111B,ROW\_1

SETB ROW\_A ;CHECKING IF KEY IS PRESSED FROM ROW\_C

SETB ROW\_B

CLR ROW\_C

SETB ROW\_D

MOV A,P3

ANL A,#00001111B

CJNE A,#00001111B,ROW\_2

SETB ROW\_A ;CHECKING IF KEY IS PRESSED FROM ROW\_D

SETB ROW\_B

SETB ROW\_C

CLR ROW\_D

MOV A,P3

ANL A,#00001111B

CJNE A,#00001111B,ROW\_3

LJMP K2

ROW\_0: MOV DPTR,#KCODE0

SJMP FIND

ROW\_1: MOV DPTR,#KCODE1

SJMP FIND

ROW\_2: MOV DPTR,#KCODE2

SJMP FIND

ROW\_3: MOV DPTR,#KCODE3

SJMP FIND

FIND: RRC A ;SEE IF CY BIT IS LOW

JNC RETURN ;IF ZERO, GET THE ASCII CODE

INC DPTR ;IF CY IS NOT ZERO, POINT TO THE NEXT COLUMN ADDRESS

SJMP FIND ;KEEP SEARCHING

RETURN: RET ;UPON RETURN, THE DPTR HOLDS THE ASCII VALUE OF THE ENTERED KEY

;SUBROUTINE TO SHOW TEXT IN LCD

TXT\_LED:CLR A

MOVC A,@A+DPTR ;COPYING THE DPTR ITEMS TO ACCUMULATOR ONE BY ONE

JZ FUNC\_END ;IF ZERO, JUMP TO END OF FUNCTION

ACALL DATAWRT

ACALL DELAY

INC DPTR

SJMP TXT\_LED

FUNC\_END:

RET

;SUBROUTINE TO CREATE NEW LINE IN THE LCD

NEW\_LINE:

MOV A,#0C0H

ACALL COMNWRT

ACALL DELAY

RET

;SUBROUTINE TO CLEAR THE LCD DISPLAY

CLEAR\_DISPLAY\_FINAL:

MOV DPTR, #CLEAR\_DISPLAY ;CLEARING DISPLAY INSTRUCTIONS ARE COPIED TO DPTR

START2: CLR A

MOVC A,@A+DPTR

ACALL COMNWRT

ACALL DELAY

JZ RETURNTO

INC DPTR

SJMP START2

LCALL DELAY

RETURNTO: RET

;COMMAND WRITE SUBROUTINE FOR GIVING COMMAND TO LCD

COMNWRT:LCALL READY ;CHECK IF THE LCD IS READY TO ACCEPT COMMAND

MOV P0, A ;COPY REG A TO PORT 0

CLR RS ;RS=0 FOR COMMAND REGISTER

CLR RW ;R/W=0 FOR WRITE

SETB E ;E=1 OF H-TO-L PULSE

LCALL DELAY ;GIVE LCD SOME TIME

CLR E ;E=0 OF H-TO-L PULSE

RET

;DATA WRITE SUBROUTINE FOR WRITNG DATA TO LCD

DATAWRT:LCALL READY ;;CHECK IF THE LCD IS READY TO ACCEPT COMMAND

MOV P0, A ;COPY REG A TO PORT0

SETB RS ;RS=1 FOR DATA REGISTER

CLR RW ;R/W=0 FOR WRITE

SETB E ;E=1 OF H-TO-L PULSE

ACALL DELAY ;GIVE LCD SOME TIME

CLR E ;E=0 OF H-TO-L PULSE

RET

;READY SUBROUTINE FOR THE LCD

READY: SETB P0.7 ;DEFINING P0.7 AS INPUT PIN FOR NOW

CLR RS ;RS=0 FOR COMMAND REGISTER

SETB RW ;R/W=1 FOR READ

WAIT: CLR E ;E=0 OF L-TO-H PULSE

LCALL DELAY ;GIVE LCD SOME TIME

SETB E ;E=1 OF L-TO-H PULSE

JB P0.7, WAIT ;AS LONG THE P0.7 BIT IS HIGH, WE WILL WAIT

RET

;DELAY SUBROUTINE

DELAY: PUSH 3 ;PUSH R3 ONTO STACK

PUSH 4 ;PUSH R4 ONTO STACK

MOV R3, #50 ;DELAY OF APPROXIMATELY 27.83 ms

HERE2: MOV R4, #255

HERE: DJNZ R4, HERE ;STAY UNTILL R4 BECOMES 0

DJNZ R3, HERE2 ;LOAD

POP 4 ;POP R4 FROM THE STACK

POP 3 ;POP R3 FROM THE STACK

RET

;HIGHER DELAY

DELAY\_H:PUSH 3 ;PUSH R3 ONTO STACK

PUSH 4 ;PUSH R4 ONTO STACK

MOV R3, #255 ;DELAY OF APPROXIMATELY 142 ms

HERE2\_: MOV R4, #255

HERE\_\_: DJNZ R4, HERE\_\_ ;STAY UNTILL R4 BECOMES 0

DJNZ R3, HERE2\_ ;LOAD

POP 4 ;POP R4 FROM THE STACK

POP 3 ;POP R3 FROM THE STACK

RET

MYLCD : DB 38H,0EH,01,06,80H,0 ;INTIALIZING LCD INSTRUCTIONS STORED HERE

CLEAR\_DISPLAY : DB 01,06,80H,0 ;CLEARING LCD INSTRUCTIONS STORED HERE

WELCOME: DB ' WELCOME! ',0 ;SHOWING WELCOME

PASSWORD: DB ' ENTER P/W:',0 ;PROMPTING TO ENTER PASSWORD

REAL\_PASSWORD: DB '5','4','5','4' ;DEFAULT PASSWORD IS THIS

DENIED: DB ' ACCESS DENIED',0 ;SHOWING DENIED

GRANTED: DB ' ACCESS GRANTED',0 ;SHOWING GRANTED PERMIT

CLOSE: DB ' ALMOST RIGHT',0 ;SHOWING ALMOST RIGHT MESSAGE

LOCKED: DB 'LOCKED',0 ;SHOWING SYSTEM IS LOCKED

RESET\_PIN: DB 'RESET P/W:4',0 ;SHOWING RESET PASSWORD OPTION

NEW\_PASSWORD: DB 'NEW P/W:',0 ;PROMPTING NEW PASSWORD

INSTRUCT\_CONTROL: DB 'ON: 5 OFF: 6',0;SHOWING MOTOR CONTROL OPTIONS

MOTOR\_CONTROL: DB ' REVERSE: 4 ',0 ;SHOWING REVERSE MOTOR OPTION

;ASCII LOOK-UP TABLE FOR EACH ROW

KCODE0: DB '1','2','3','A' ;ROW 0

KCODE1: DB '4','5','6','B' ;ROW 1

KCODE2: DB '7','8','9','C' ;ROW 2

KCODE3: DB '\*','0','#','D' ;ROW 3

END