ORG 00H

RS EQU P1.7

RW EQU P1.6

E EQU P1.5

KEYPORT EQU P3

LCDPORT EQU P0

MOV SP,#70H

MOV PSW,#00H

START:SETB P1.0 ; initially turns off the LED

SETB P1.1 ; initially turns off the buzzer

MOV R0,#30H ; where the ASCII is to be stored

MOV R5,#03 ; how many characters to take input

ACALL LCD\_INI ; initializes the LCD

MOV DPTR,#WELCOME1 ; moves DPTR to the location of WELCOME1

ACALL STRPRNT ; prints out the string

MOV A,#0C0H

ACALL COMNWRT ; moves cursor to next line

ACALL DELAY\_1

MOV DPTR,#WELCOME2 ; moves DPTR to the location of WELCOME2

ACALL STRPRNT

ACALL KEYPAD ; taking input from the keypad

ACALL BCD\_C ; converts that to BCD

ACALL LCD\_INI ; clears the LCD

MOV DPTR,#ROL ; moves DPTR to the location of ROL

CLR A

ACALL STRPRNT ; prints out the string

MOV R5,#1 ; only 1 character to take input

MOV R0,#33H ; ASCII to be stored here

ACALL KEYPAD ; taking input from the keypad

ACALL LCD\_INI ; clearing the LCD

MOV DPTR,#SPEED ; moves DPTR to the location of SPEED

CLR A

ACALL STRPRNT ; prints out the string

MOV A,#0C0H

ACALL COMNWRT ; moves cursor to next line

ACALL DELAY\_1

MOV DPTR,#SPEED\_C ; moves DPTR to the location of SPEED\_C

CLR A

ACALL STRPRNT ; moves cursor to next line

MOV R5,#1 ; only 1 input character

MOV R0,#34H ; ASCII storing location

ACALL KEYPAD ; taking input from keypad

ACALL LCD\_INI ; clearing the LCD

CLR A

MOV DPTR,#ROTATING ; moves DPTR to the location of ROTATING

ACALL STRPRNT ; prints out the string

CLR P1.0 ;turns on the LED

ACALL STEP\_C ; calculates the number of steps required to move the

specified angle

ACALL SPINMF ; spins according to the number of steps

CLR P1.1 ; turns on buzzer

ACALL DELAY\_B ; delay to control the duration of buzzer beep

SETB P1.1 ; turns off buzzer

SETB P1.0 ; turns off the LED

ACALL LCD\_INI ; clears the LCD

LJMP START ; then jumps to start

KEYPAD: ; subroutine to take input from keypad

E1: MOV A, #0FH

MOV KEYPORT, A

K1: MOV KEYPORT, #00001111B

MOV A, KEYPORT

ANL A, #00001111B

CJNE A, #00001111B, K1

K2: ACALL DELAY\_1

MOV A, KEYPORT

ANL A, #00001111B

CJNE A, #00001111B, OVER

SJMP K2

OVER: ACALL DELAY\_1

MOV A, KEYPORT

ANL A, #00001111B

CJNE A, #00001111B, OVER1

SJMP K2

OVER1: MOV KEYPORT, #11101111B

MOV A, KEYPORT

ANL A, #00001111B

CJNE A, #00001111B, ROW\_0

MOV KEYPORT, #11011111B

MOV A, KEYPORT

ANL A, #00001111B

CJNE A, #00001111B, ROW\_1

MOV KEYPORT, #10111111B

MOV A, KEYPORT

ANL A, #00001111B

CJNE A, #00001111B, ROW\_2

MOV KEYPORT, #01111111B

MOV A, KEYPORT

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

FIND: RRC A

JNC MATCH

INC DPTR

SJMP FIND

MATCH:CLR A

MOVC A,@A+DPTR

ACALL DATAWRT

ACALL DATASV

ACALL DELAY\_1

DJNZ R5,E1

RET

BCD\_C:MOV A,30H ; subroutine to convert the number to BCD

SUBB A,#30H

MOV B,#100D

MUL AB

MOV 40H,A

MOV 41H,B

MOV A,31H

SUBB A,#30H

MOV B,#10D

MUL AB

MOV 42H,A

MOV A,32H

SUBB A,#30H

MOV 43H,A

ACALL ADDER

RET

STEP\_C:MOV R0,45H ; subroutine to calculate the number of steps

required

MOV R1,41H

MOV R2,#0AH

MOV R3,#0

LCALL T\_DIV

MOV 55H,R4

MOV 56H,R5

LCALL T\_MUL

MOV 57H,R4

MOV 58H,R5

RET

SPINMF:MOV R1,57H ; subroutine to spin motor according to step

number

MOV R2,58H

MOV R3,#0

MOV R4,#0

MOV 70H,#66H

LOOP\_S:CLR C

MOV A,70H

MOV R7,33H

CJNE R7,#'>',LEFT

RR A

SJMP CON

LEFT: RL A

CON: MOV 70H,A

MOV P2,A

MOV A,34H

CJNE A,#'L',D\_2

ACALL DELAY\_3

SJMP CT

D\_2: ACALL DELAY\_2

CT: MOV A,R3

ADD A,#01

MOV R3,A

JNC CONT

INC R4

CLR C

CONT: MOV A,R3

MOV B,R1

CJNE A,B,LOOP\_S

NEXT: MOV A,R4

MOV B,R2

CJNE A,B,LOOP\_S

RET

ADDER:CLR C ; adder subroutine used for STEP\_C subroutine

MOV A,42H

MOV B,43H

ADD A,B

MOV 44H,A

ACALL T\_SUM

RET

STRPRNT:CLR A ; subroutine used to print out string

MOVC A,@A+DPTR

JZ BR

ACALL DATAWRT

ACALL DELAY\_1

INC DPTR

SJMP STRPRNT

BR: RET

COMNWRT: ; LCD command subroutine

MOV B, A

ANL A, #0F0H

MOV LCDPORT, A

CLR RS

CLR RW

SETB E

ACALL DELAY\_1

CLR E

MOV A, B

SWAP A

ANL A, #0F0H

MOV LCDPORT, A

SETB E

ACALL DELAY\_1

CLR E

MOV A, B

ACALL DELAY\_1

RET

DATAWRT: ; LCD write subroutine

MOV B, A

ANL A, #0F0H

MOV LCDPORT, A

SETB RS

CLR RW

SETB E

ACALL DELAY\_1

CLR E

MOV A, B

SWAP A

ANL A, #0F0H

MOV LCDPORT, A

SETB E

ACALL DELAY\_1

CLR E

MOV A, B

ACALL DELAY\_1

RET

DATASV:MOV @R0,A ; subroutine to save data to RAM location

INC R0

RET

;LCD initialization subroutine

LCD\_INI:MOV A, #02H ; 4 bit mode

ACALL COMNWRT

MOV A,#28H ; 4 bit mode initialization

ACALL COMNWRT

ACALL DELAY\_1

MOV A,#0CH

ACALL COMNWRT ; display on, cursor blinking

ACALL DELAY\_1

MOV A,#06H ; move cursor to the right

ACALL COMNWRT

ACALL DELAY\_1

MOV A,#01H ; clear screen

ACALL COMNWRT

ACALL DELAY\_1

MOV A,#80H ; force cursor to the begining of 1st line

ACALL COMNWRT

ACALL DELAY\_1

RET

T\_DIV: ; 16 bit division subroutine

CLR C

MOV R4,#00h

MOV R5,#00h

MOV B,#00h

div1:

INC B

MOV A,R2

RLC A

MOV R2,A

MOV A,R3

RLC A

MOV R3,A

JNC div1

div2:

MOV A,R3

RRC A

MOV R3,A

MOV A,R2

RRC A

MOV R2,A

CLR C

MOV 07h,R1

MOV 06h,R0

MOV A,R0

SUBB A,R2

MOV R0,A

MOV A,R1

SUBB A,R3

MOV R1,A

JNC div3

MOV R1,07h

MOV R0,06h

div3:

CPL C

MOV A,R4

RLC A

MOV R4,A

MOV A,R5

RLC A

MOV R5,A

DJNZ B,div2

MOV R3,05H

MOV R2,04H

RET

T\_MUL:MOV R0,56H ;MSB1 ; 16 bit multiplication subroutine

MOV R1,#0 ;MSB2

MOV R2,55H ;LSB1

MOV R3,#57 ;LSB2

MOV A,R2

MOV B,R3

MUL AB

MOV R4,A

MOV R5,B

MOV A,R2

MOV B,R1

MUL AB

MOV R6,B

ADDC A,R5

MOV R5,A

MOV A,R1

MOV B,R4

MUL AB

ADDC A,R5

MOV R5,A

MOV A,B

ADDC A,R6

MOV R6,A

MOV A,R1

MOV B,R2

MUL AB

ADDC A,R6

MOV R6,A

MOV A,B

ADDC A,#00H

MOV R7,A

RET

T\_SUM:CLR C ; subroutine used along with BCD\_C subroutine

MOV A,40H

ADD A,44H

MOV 45H,A

JC INCRE

RET

INCRE:INC 41H

RET

DELAY\_1:SETB PSW.3

MOV R3, #50

Y0: MOV R1, #255

Y1: DJNZ R1, Y1

DJNZ R3, Y0

CLR PSW.3

RET

DELAY\_2:MOV R6, #010H

L0: MOV R5, #0FFH

L1: DJNZ R5, L1

DJNZ R6, L0

RET

DELAY\_3:MOV R6, #028H

N0: MOV R5, #0FFH

N1: DJNZ R5, N1

DJNZ R6, N0

RET

DELAY\_B:MOV R7, #4H

pp00: MOV R6, #0FFH

pp0: MOV R5, #0FFH

pp1: DJNZ R5, pp1

DJNZ R6, pp0

DJNZ R7, pp00

RET

CLR\_LCD:MOV A,#01H ; subroutine to clear LCD

ACALL COMNWRT

ACALL DELAY\_1

RET

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

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

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

KCODE3: DB '<','0','>','N' ;ROW 3

WELCOME1: DB '3 DIGIT INPUT',0 ;first line of 1st MSG to print

WELCOME2: DB '(000-999):',0 ;second line of 1st MSG to print

ROL: DB 'ROTATE CW/CCW? ',0 ;second MSG to print

ROTATING: DB 'ROTATING...',0 ;third MSG to print

SPEED: DB 'SELECT SPEED:',0 ;1st line of forth MSG to print

SPEED\_C: DB 'L:LOW H:HIGH ~ ',0 ;2nd line of forth MSG to print

END