**;Convert 17H to 71H**

MOV A,#17H

SWAP A

;**To Load data to Accumulator**

MOV A, #05H

**;Load A with 60h, load Register R1 with FFH and then Exchange the data between both the Register**

MOV A,#60H

MOV R1,#0FFH

XCH A,R1

**;To Load decimal to a Register and then Transfer it to another Register**

MOV R0,#0CH

MOV A,R0

MOV R7,A

**;Perform Division between two 8 bit numbers and then perform 2's complement of the remainder**

MOV A, #06H

MOV B, #05H

DIV AB

MOV A,B

CPL A

INC A

**;To ADD two 8-bit numbers**

MOV A,#06H

ADD A,#05H

;**To DIVIDE two 8-bit numbers**

MOV A,#06H

MOV B,#05H

DIV AB

**;To MULTIPLY two 8-bit numbers**

MOV A,#06H

MOV B,#05H

MUL AB

**;To SUBTRACT two 8-bit numbers**

MOV A,#06H

SUBB A,#05H

**;Perform 05H AND 06H OR F0H**

MOV A, #05H

ANL A, #06H

ORL A, #0F0H

**; Perform FFH XOR (NOT 08H)**

MOV A, #08H

CPL A

XRL A, #0FFH

**;To convert AAH to 55H**

MOV A,#0AAH

RL A

**;To Mask Lower nibble of a 8-but number**

MOV A,#06H

ANL A,#0F0H

;LOWER NIBBLE MASKING USE F0H

;UPPER NIBBLE MASKING USE 0FH

**;To perform 2'S Compliment of an 8-bit numbers**

MOV A,#06H

CPL A

INC A

**;To perform AND operation between two 8-bit numbers**

MOV A,#06H

ANL A,#05H

**;To perform OR operation between two 8-bit numbers**

MOV A,#06H

ORL A,#05H

**;To perform XOR operation between two 8-bit numbers**

MOV A,#06H

XRL A,#05H

**; Perform Subtraction of two 8-bit numbers and store result in the R0 Register of R2 Register Bank**

0000| MOV R1,#0AH

0002| SUBB A,#04H

0004| SETB RS1

0006| MOV R0,A

**; Perform addition of two 8-bit numbers and store the result in R0 Register of Register Bank1**

MOV A, #0AH

ADD A, #04H

SETB RS0

MOV R0,A

**; Perform addition of two 8-bit numbers and stored in R1 Register and R2 Register of Bank 1 and store the result in Ro Register of Register Bank R1**

0000| SETB RS0

0002| MOV R1,#0AH

0004| MOV R2,#04H

0006| MOV A,R1

0007| ADD A,R2

0008| MOV R0,A

**;Perform Multiplication between two 8-bit numbers and store the result in R0 Register of Register Bank 3**

MOV A,#0AH

MOV B,#04H

MUL AB

SETB RS0

SETB RS1

MOV R0,A

**;Add two 8-bit numbers with carry adjustment**

MOV B, #00H

MOV A, #0FFH

ADD A, #0FEH

JNC L1

INC B

L1: MOV R1,A

MOV R2,B

**;Add two 8-bit numbers that generate an Auxillary carry and then perform Logical OR operation between Carry and Auxillary carry**

MOV A, #0EH

ADD A, #16H

MOV R1,PSW

ORL C, AC

**;Add two 8-bit numbers that generate carry and store carry in R1 Register of Register BAnk 3 and then complement the carry flag.**

MOV A, #0FFH

ADD A, #0FEH

SETB RS1

SETB RS0

MOV R0,A

MOV R1,PSW

CPL C

NOP

**;Find 2's Complimnet of 16 bit number**

MOV R0,#00H

MOV A, #00H

CPL A

ADD A,#01H

JNC L1

INC R0

L1: MOV R1,A

MOV A,# 40H

CPL A

ADD A,R0

MOV R2,A

**;TO ON AND OFF A PATTERN OF 8 LED**

MOV A, #00H

MOV P1,A

ACALL DELAY

MOV A,#0FFH

MOV P1,A

ACALL DELAY

DELAY: MOV R0,#02H

L1: DJNZ R0,L1

RET

**;DESIGN AN DOWNCOUNTER FROM 8 TO 0**

;MOV P1, #01111111B

;MOV P1, #00111111B

;MOV P1, #00011111B

;MOV P1, #00001111B

;MOV P1, #00000111B

;MOV P1, #00000011B

;MOV P1, #00000001B

;MOV P1, #00000000B

MOV R0,#08H

MOV A,#08H

L1:MOV P1,A

ACALL DELAY

DEC A

DJNZ R0,L1

DELAY: MOV R1,#02H

L2:DJNZ R1,L2

RET

**;DESIGN AN UPCOUNTER FROM 0 TO 8**

;MOV P1, #11111110B

;MOV P1, #11111100B

;MOV P1, #11111000B

;MOV P1, #11110000B

;MOV P1, #11100000B

;MOV P1, #11000000B

;MOV P1, #10000000B

;MOV P1, #00000000B

HERE:MOV R0,#08H

MOV A,#00H

L1:MOV P1,A

ACALL DELAY

INC A

DJNZ R0,L1

SJMP HERE

DELAY: MOV R1,#02H

L2:DJNZ R1,L2

RET

**;ON D0 TO D7 BIT INDIVIDUALLY AND CONTINUOUSLY**

;HERE:MOV P1, #11111110B

;MOV P1, #11111101B

;MOV P1, #11111011B

;MOV P1, #11110111B

;MOV P1, #11101111B

;MOV P1, #11011111B

;MOV P1, #101111111B

;MOV P1, #01111111B

;SJMP HERE

HERE:MOV P1,#0FEH

ACALL DELAY

MOV P1,#0FDH

ACALL DELAY

MOV P1,#0FBH

ACALL DELAY

MOV P1,#0F7H

ACALL DELAY

MOV P1,#0EFH

ACALL DELAY

MOV P1,#0DFH

ACALL DELAY

MOV P1,#0BFH

ACALL DELAY

MOV P1,#7FH

ACALL DELAY

SJMP HERE

DELAY:MOV R0,#02H

L1:DJNZ R0,L1

RET

**;TO GLOW A PATTERN OF 8 LED**

MOV A, #00H

MOV P1,A

**;TO ON AND OFF A PATTERN OF 8 LED CONTINUOUSLY**

HERE:MOV A, #00H

MOV P1,A

ACALL DELAY

MOV A,#0FFH

MOV P1,A

ACALL DELAY

SJMP HERE

DELAY: MOV R0,#02H

L1: DJNZ R0,L1

RET

**;TO ON AND OFF A PATTERN OF 8 LED**

MOV A, #00H

MOV P1,A

ACALL DELAY

MOV A,#0FFH

MOV P1,A

ACALL DELAY

DELAY: MOV R0,#02H

L1: DJNZ R0,L1

RET

**;TO ON AND OFF LOWER NIBBLE AND UPPER NIBBLE OF A PATTERN OF 8 LED**

MOV P1,#0F0H;MOV P1,#11110000B

ACALL DELAY

MOV P1,#0FH;MOV P1,#00001111B

ACALL DELAY

DELAY: MOV R0, 0FFH;MOV P1,#00000001B

RET

**To ADD two 8-bit numbers and display in LED**

MOV A,#06H

ADD A,#05H

MOV A, #0BH

MOV P1,A

**To ADD two 8-bit numbers and display on LED**

MOV A,#06H

ADD A,#05H

HERE:SETB P0.7

CLR P3.3

CLR P3.4

MOV P1,#0f9H

SETB P0.7

SETB P3.3

CLR P3.4

MOV P1,#0f9H

SJMP HERE

**To ADD two 8-bit numbers and display in LCD display**

MOV A,#06H

ADD A,#05H

MOV A, #38H; To intialize 16x2 LCD

ACALL COMNWRT

ACALL DELAY

MOV A, #0EH; Display on, Cursor blinking

ACALL COMNWRT

ACALL DELAY

MOV A, #01H; Clear display screen

ACALL COMNWRT

ACALL DELAY

MOV A, #06H; Increment Cursor

ACALL COMNWRT

ACALL DELAY

MOV A, #84H; 4th Position 1st Line

ACALL COMNWRT

ACALL DELAY

MOV A, #'E'

ACALL DATAWRT

ACALL DELAY

MOV A, #'L'

ACALL DATAWRT

ACALL DELAY

MOV A, #'E'

ACALL DATAWRT

ACALL DELAY

MOV A, #'V'

ACALL DATAWRT

ACALL DELAY

MOV A, #'E'

ACALL DATAWRT

ACALL DELAY

MOV A, #'N'

ACALL DATAWRT

ACALL DELAY

HERE:SJMP HERE

COMNWRT: MOV P1,A

CLR P3.5

SETB P3.7

ACALL DELAY

CLR P3.7

RET

DATAWRT:MOV P1,A

SETB P3.5

SETB P3.7

ACALL DELAY

CLR P3.7

RET

DELAY: MOV R1,#0FFH

L1:DJNZ R1,L1

RET